T—Y—N—E—&—W—E—A—R

# TYNE & WEAR

# ATARI 8-BIT USER GROUP

Newsletter of TWAUG

Software
Editorial
Buy & Sell
Hardware

TWAUG 8-BIT

Reviews
Help line
Section
Repair Info

Public Domain Library

ISSUE #5

SEPTEMBER/OCTOBER 1993

U—S—E—R—G—R—O—U—P

# BRING YOUR EIGHT UP TO DATE with power products from COMPUTER SOFTWARE SERVICES

## THE BLACK BOX

The BLACK BOX is an add-on board for the Atari 600XL,800XL and 130XE 8-bit computers. It is a T-shaped board that plugs into the PBI port of the XL computer, or the ECI and cartridge ports of the 130XE. Connectors for both types of computers are built into the BLACK BOX so no adapter boards are necessary. A cartridge port is available on the board itself for 130XE users.

The BLACK BOX provides many unique and useful functions. The four primary functions are:-

* RS-232 serial modem port
* Parallel printer port
* SASI/SCSI hard disk port
* Operating System enhancements

The BLACK BOX is $199.95 for the basic unit, and $249.95 with an onboard 64K printer buffer.
Shipping and Handling extra.

## THE BLACK BOX ENHANCER

A must for all BLACK BOX owners. The BLACK BOX ENHANCER is a plug-in module for your BLACK BOX, enhancing the printer functions and adding an instantly available, full featured sector editor!

Installation of the BLACK BOX ENHANCER requires one simple solder connection. Only $49.95 plus shipping/handling.

## THE FLOPPY BOARD

Our latest and greatest product. The FLOPPY BOARD is an add-on expansion board for the BLACK BOX interface. It allows the use of the same inexpensive floppy drive mechanisms used in IBM computers. The FLOPPY BOARD is the first floppy drive interface to support "high density" floppy drive mechanisms in either 5.25 inch or 3.5 inch. Built into the FLOPPY BOARD are our BLACK BOX ENHANCER and a version of our SUPER ARCHIVER to allow copying of protected disks for 3.5 inch format. Included with the FLOPPY BOARD is our program to read and write to IBM or ST formatted disks. This makes the FLOPPY BOARD the best way to transfer files to and from your 8-bit.

The FLOPPY BOARD is only $149.95 plus shipping & handling.

## THE MULTIPLEXER

This device brings the power and flexibility of larger systems to your 8-bit. The Multiplexer is a collection of cartridge interface boards that allow up to 8 Ataris to read and write to the same drives (typically a hard disk), access the same printer(s), and talk to each other. It is the first practical networking system for the Atari 8-bit computer.

One "master" computer (any 8-bit) is equipped with the master Multiplexer interface. Then up to 8 "slave" computers hook up to this master, each having their own slave interface.

The "common" peripherals (things that are to be shared) are connected to the master. On each slave, all disk and printer I/O is routed through the master, so no extra disk drives are needed.

The Multiplexer sells for $199.95 for a master and two slave units with cable. Additional slave units are $89.95 each, plus shipping/handling.

## THE SUPER ARCHIVER II

The SUPER ARCHIVER II edits and copies all enhanced density programs plus retains all the features of the SUPER ARCHIVER.

The SUPER ARCHIVER II is only $99.95 plus shipping & handling. NOTICE: if you already have THE SUPER ARCHIVER you may upgrade to S.A.II for only $29.95 plus shipping/handling. Software only.

## THE BIT WRITER

The Super Archiver BIT WRITER is capable of duplicating even the "uncopyable" Electronic Arts and Synapse Syn-series, which employ 34 full sector tracks. The BIT WRITER must be used with the SUPER ARCHIVER

The BIT WRITER is only $79.95 plus shipping/handling.

## THE ULTRA SPEED PLUS OS

The Operating System that should be in every XL/XE computer! The Ultra Speed Plus puts unbelievable speed and convenience at your fingertips.

Use any DOS to place Ultra Speed formats on your disks (with XF551 or modified 1050 drives), reading and writing at this speed with most programs. This high speed mode can be turned off for maximum compatibility.

Four simple solder connections are required for installation if your machine has a socketed OS ROM. The Ultra Speed OS is only $69.95 plus shipping/handling.

For more information on these and other 8-bit products:

**CONTACT**
**COMPUTER SOFTWARE SERVICES**
**PO BOX 17660**
**ROCHESTER, NEW YORK 14617**
**USA**

**ORDERING LINE: (716) 429-5639**
**FAX: (716) 247-7158**
**BBS: (716) 247-7157**

or contact T.W.A.U.G. we will do our best to help.

# TWAUG NEWSLETTER

## EDITORIAL

Who to blame!!!

John Matthewson
David Ewens
Max Gerum

Well here we are again and with a much thicker issue, we had to include extra four pages to accomodate the extra material we received for this issue.

With the A.M.S.7 show coming up we thought it appropriate to include one or two extra adverts relevent to this all micro show.

I am afraid at least two material contributors will be dissapointed this time round, but we promise you that your articles will be published in the next issue. The articles are already printed out ready for issue #6.

We regret to inform you that one regular article is missing, the BASIC TUTORIAL, we are sorry about it. It is not our fault we haven't received it yet, not up to the publication of this issue. We have no idea what has happened to the material, it could be the mail or the author has been unable to post it to us in time. It has to come from Israel and we believe the writer is in the forces. We are hoping that we will be able to continue with this tutorial in the next issue.

T.W.A.U.G. wishes to inform you that we have a stand at the A.M.S.7 at Stafford on Saturday the 13th of November, and we hope to see you there. See the advert on the back page.

The next issue will be ready by mid-November.

## CONTENTS


EDITORIAL 3

NEW! PROGRAMMER'S CLUB
Need help programming? 4

ADVERT FOR NEW SOFTWARE
Check page 5

TEXTPRO PERFECT WORDPROCESSOR
by Max Gerum 7

LETTER SECTION 12

MARK'S GAMES COLUMN
by Mark Stinson 14

THE BLACK BOX
The promised article 16

COMPUTER EXPERIENCE
by Bill Jackson 17

HINTS & TIPS
by David Ewens 18

FIVE LINERS 19

VERTICAL BLANK INTERRUPT
by Nir Darey 21

BUY, SALE & CONTACT
and a TEN LINER 22

DIY & 1050 DRIVE
by John Foskett 23

WHAT NO DISK CONTENT?
an apology by Max 24

CRACKING THE CODE
Machine code for beginner 25

REVIEWS
by Mark Fenwick 29

REVIEW
by Tom Wood 31

DISK CONTENT 32

GERMANY'S ATARI 8-BIT SHOW
Report by Derek Fern 33

A.M.S.7 ADVERT 36

**NEW for the 8-bit Atari...**

The Newsletter of
The Atari Classic Programmer's Club

The lack of commercial quality software for the greatest of 8-bits is depressing at the best of times. For too long, we've had to suffer the injustices given to the Atari community by some of the larger software houses, particularly in the UK. So, instead of wringing our hands and watching from the sidelines, why don't you get up and DO SOMETHING ABOUT IT!!

When you join The Atari Classic Programmer's Club, you will be eligible for assistance with any aspect ot writing a piece of software on the Atari 8-bit. We can help by providing graphics, Sound, Music, PMGs, UDGs, Play Testing, programmer's helpline, and possible publication of your finished software. We are also happy to help people who have just taken up programming or have just moved to a new language.

We have membership fees to cover 6 months, 12 months and Life memberships, so anybody can join: For full details, send large SAE to:

ACPC, Pen-Tyddyn, Capel Coch, LLangefni, Anglesey, Gwynedd LL77 7UR, Wales.

**Everything the Atari programmer needs.**

---

D.G.S. Dean Garraghty's Software.

62 Thomson Ave., Balby, Doncaster. DN4 0NU. TEL. (0302) 855026.

Not only does Dean produce a very good News-Disk, but he also runs an excelent PD library. He has recently branched out into the commercial side of the 8-BIT software scene with the new game called MINE SWEEPER, and more recently the very good new QUICK language.

Why not drop him a line, or give him a phone call and ask him to send you his latest D.G.S catalogue. If you are going to the AMS7 show at Stafford on the 13th of November, why not visit him at his stand and see what he has to offer.

---

**COMING SOON...**

# MENU PRINT ELITE

version 1.0

Remove all those disk cataloguing hassles with MENU PRINT ELITE, the complete Atari 8-bit disk cataloguing system. Features include:

- Ability to read a large number of menu and DOS systems including Rob C, Multiboot, DOS 2.0, DOS 2.5, DOS 4, DOS XL, SpartaDOS, Howfen Menu, Howfen DOS, Transdisk and others..
- Create and print labels.
- Assign individual disk numbers for easy storage of disks.
- Additional user input feature for cataloguing of unrecognisable disks.
- Dump catalogue to disk for printing out later.
- And much, much more ...

So, if you've got a large collection of disks but don't know what's on them, Menu Print Elite will find out for you.

Release Date: November 1993  Price £9.95

# ACPC

The Atari Classic Programmer's Club.

# MICRO DISCOUNT

Here is some good news for all 8-bitters, MICRO DISCOUNT will release two New Hardware kits and twelve New Software Titles at the A.M.S.7 show.

NEW HARDWARE

STEREO CONVERSION KITS
NEW SOUND REPLAY CARTRIDGE

NEW SOFTWARE

" THE BRUNDLES "
STEREO DRUM EDITOR "8 CHANNEL DRUM MACHINE"
DRACONUS and ZYBEX both in "STEREO SOUND"
16 BIT SOUND PLAYER DISK
OPERATION BLOOD "LIGHT GUN VERSION"
IMAGINE  BANK BANG
SPECIAL FORCES  WHEEL OF FORTUNE
TANKS  THE MERCENARY
ARTIFACT

## YORKIE 256K plug-in memory upgrade.

Yes it's available once again, but in limited quantities.

The Yorky simply plugs into the PBI on the back of your 800XL (or internally upgraded to 64k, 600XL) to give you 256k of RAMBO XL compatible bank switched memory. NO soldering is required and you don't have to open up your machine.

It comes complete with a printed manual and a disk full of software designed to make use of the extra memory.

Price 50 pounds + 2 pounds p&p.

For enquiries and orders please write to:-

RICHARD GORE, 79 SPROTBROUGH ROAD, SPROTBROUGH, DONCASTER, DN5 8BW

or telephone me (Richard) on (0302) 784642 any weekend between 7pm Friday and 4pm Sunday.

NB. I can only guarantee this upgrade to work on computers that use a UK standard power supply.

## PHOENIX.

The new disk based news letter from Ireland, produced by Robert Paden.

Robert recently had to give up producing the I.A.U newsletter after fur issues because of lack of support.

PHOENIX will be a double sided disk, side 'A' will be packed full of text files containing Articles, reviews and much much more. Side 'B' will contain a good selection of PD software.

The first 6 issues will be available from either the TWAUG, or from New Atari User PD libraries at £2.50 per issue.

If PHOENIX proves to be a success (which we hope it will), then from issue 7, it will only be available from Robert Paden himself.

Issue one of PHOENIX will hopefully be available by the end of August, or sometime in September. Why not give it a try.

## MICRO DISCOUNT

Now the largest Mail Order Stockist of Atari 8 Bit Hardware & Software in the U.K.

New Software for 1993

The last Guardian
Tagalon Disks £5.95 H & S Speed Tapes £4.95
Adax Disk £5.95
Eureka Disk £5.95
Guitar Wizard Disk £10.95
Hans Kloss Disk £5.95
and many more

Hardware & Upgrade Kits

Atari Disk drives (re-cond) £100.00
Data Recorders XC12 (re-cond)
Standard £28.00 Turbo £43.00
Upgrades from £18.95 to £51.95
and much, much more

To receive a 32 page catalogue you must first register your name and full postal address on our Data Base. The cost of this service is just £3.50 for 6 copies (1 every 8 weeks): £6.00 Europe: £10.00 Elsewhere. Prices on all items unless stated do not include post and packing, therefore please contact before ordering with details of your requirements to confirm availability and postage. Allow 28 days for delivery.

MICRO DISCOUNT
265 Chester Road
STREETLY
West Midlands B74 3EA
Tel:021-353-5730 or Fax:021-352-1669

## NEW ATARI USER PAGE 6

The only magazine left in this country that supports the 8-bit

There is a large Public Domain Library available. Your support is needed to keep the magazine going.

It is now only available by subscription, for more details write to:

PAGE 6
P.O.BOX 54
STAFFORD ST16 1TB

# TEXTPRO WORD PROCESSOR

by Max Gerum

In the last issue I told you that I do all my printing from the RAMdisk and subdirectory. DOS 2.5 doesn't support subdirectories, and as far as I know you can only have two RAMs of 707 sectors each, that is if it has been upgraded. Anyone who needs a DOS that supports a RAMdisk of 256K, 320K and 1 MEG plus subdirectories, should get SpartaDOS or MyDOS. I cannot comment on Sparta X as I haven't got it, neither will I comment on disk based Sparta 3.2, I'm not using it.

Before I continue writing about TextPro I would like to explain how I load TP from the RAMdisk. When running TP from the RAMdisk you need to make a small modification to the program itself. The change that has to be made is for the supporting files to run from the RAMdisk also. If you haven't already made the changes and would like to do so but don't know where to start, why not get the RAMdisk Patcher. It is called TPRAMDSK.BAS and is available from our PD library with TextPro. This program will do the necessary changes for you to run TP from D9: or whatever you want to call your RAMdisk. First make sure your RAMdisk is formatted, then go into basic and run the Patcher program. The program will prompt you to enter the filename of your TextPro, it will then load TP into memory. The Patcher will not change your original file of TP on the disk, the changes only occur in the memory. Type in the RAMdisk number when prompted, and before writing the modified file to RAMdisk the program also gives you the option to change the destination filename, or you can use the default filename which is displayed.

Why not copy the modified TextPro from the RAMdisk to the original disk or make another working disk to run entirely from D8:. But do remember the supporting programs must also be copied to the RAMdisk with TextPro before running the program. The supporting files that load automatically when you boot TP are "TEXTPRO.MAC, TEXTPRO.CNF, and TEXTPRO.FNT, these three files must be copied to D8:. There are a couple of programs available for copying these files mentioned above to your RAMdisk automatically.

Now let us continue with TextPro, in the last issue I explained the main commands, now lets take a look at some more commands and what else TextPro can offer.

All word processors seem to have more or less the same amount of space in the text buffer and that's too little. Have you ever loaded a very large document into a word processor and when you got to the end found a large part was missing? TextPro has the ability to segment any size file, and this option is built right into the program. With TP you can pick the size of each segment from the editor and place the breaks where you want them before segmenting the file.

Use the ",C" Parameter command. Let me explain how to divide a large file for editing. Simply load a large file into the editor. If the file is too large for the text buffer, it will stop loading with 80 characters left in the editor. A message will be displayed in the command line thus "No Errors, Linked Load". Using the CTRL+E command you can move to the end of the loaded file and place the cursor at the spot where you want the file segmented. And to find the exact size of the segment that you are selecting use CTRL-U, that's the Used and Unused memory command. The number of bytes used and unused and the exact cursor position of the selected segmented file will be displayed in the command line.

TEXTPRO WORD PROCESSOR continued

Now we must save this first segment to disk. Just use the CTRL-S (save file) command, specify the drive number and filename you want for the segment. The only difference is that you add after the filename the ",C" parameter command. This parameter will specify a save from the top of the file to the cursor position.

The next thing to do is make more room in the editor. Since you have saved the segment above the cursor as a separate file, we don't need to keep it any longer in the editor. Use the SELECT+CTRL-U (delete to top of file) command, this deletes the already saved segment.

Now use the CTRL-E command again and the cursor will be placed to the exact spot where the previous load to the editor ended. It could be right in the middle of a word, to append the disk file to the editor from this exact spot use the CTRL-L command. The command line will now turn green indicating that an appended load is being selected. Use the same disk number and filename as the original file. To make it easier for you press the SELECT-P or p combination, this will print the disk number plus original filename to the command line.

This would normally load an appended file from the start of the disk file, what you really want is to append the load from the position the previous load stopped. To select this add the ",C" parameter to the filename in the command line, simply type ,C do not include a space. The file will now load and append to the editor from the spot where the original file ended and continue to the full size of the buffer.

You may repeat this segmenting, clearing and loading procedure as many times as needed to segment an entire file exactly as you wish and save the files to disk under any filenames you desire. When the file is completely loaded to the editor the message "No Errors" will be displayed in the command line.

You use this method mentioned with a version of TextPro that hasn't any banks, like TextPro version 4.54. If you are using TextPro version 4.56 you have three BANKs available to do the segmentation in. All three banks combined are over 40K in size enough room to accommodate most large files.

TextPro version 4.56 is available from our PD library, it is set up as an autorun system, and the Banks are activated automatically when booted. Let me explain how to load a file that is too large for one bank to accommodate. First load the macro file "XLOAD.MAX", to load a .MAX file you just press start and type the filename but not the extender, the extender .MAX is added by the program. A prompt will be displayed in the command line, you only need to answer with the key press 'Y' followed by the filename of the disk file. When the first bank is loaded you again answer the prompt by typing 'Y' and the same for bank two. The BANKs are numbered with 'M' for main, '1' for Bank 1, and '2' for Bank 2. To move from the main Bank to Bank 1 press SELECT+CTRL-B, from Bank 1 to Bank '2' press CTRL-B and the same key press to return to Bank 1. But to return from Bank 2 to the Main Bank you press SELECT+CTRL-B.

Each bank can be edited seperately and saved, or edited and saved back to disk as one file. You can use the macro file XSAVE.MAX to append the files, it does the appending automatically, instead of using the ",C" parameter explained above.

TEXTPRO WORD PROCESSOR continued

You can also print from the three banks and append the printing using the <goto> command. Go to the end of the file in the Main bank and type 'g' in inverse, to print in inverse hold down SELECT and type the lower case 'g' and add a 1 without holding down SELECT. Go into Bank 1 and add a 'g' and 2 to the end of that text if a file extends to Bank 2.

The segment function of TextPro is one major feature I do like, there aren't many, if any, word processors with this option. Now let me explain some more additional editing commands of TextPro. The following commands are for moving the cursor quickly around the editing screen.

SHIFT-LEFT/RIGHT arrow moves the cursor to the beginning of the previous word or the next word respectively.

SHIFT-UP/DOWN arrow moves the cursor to the beginning of the previous paragraph or next paragraph respectively.

CTRL-, and CTRL-. moves the cursor to the beginning of the previous sentence or the next sentence respectively.

CTRL-Q moves the cursor to the beginning of the screen line.

CTRL-Z moves the cursor to the end of the screen line.

CTRL-H <home cursor> command pressed once the cursor goes to the top of the screen. If held down or pressed twice in succession the cursor jumps to the top of the file.

CTRL-E <end of file> command moves the cursor to the end of the file.

I am sure every body knows about the Insert and Replace modes and what these modes do. In the insert mode text is inserted at the cursor and the remainder of the file after the cursor is moved to accommodate the new text. In the replace or overwrite mode the new text overwrites the previous text. These modes are toggle by the

CTRL-I <insert toggle> command when you press CTRL-I the mode you have just entered will be indicated in the command line. The command line will be blue in the insert mode and black in the replace mode.

There may come a time when you have need to insert some text in the middle of a large file. When editing, the Insert mode can slow the editor down, because all the characters after the cursor need to be moved to make room for the new text. This can also cause the editor to DROP some characters since the editor hasn't had time to complete the insertion before the next key is pressed.

For this reason the Replace mode should be used, in this mode the characters would not be moved. The disadvantage to this method is that the file must be opened up to make room for the additional text, otherwise you would overwrite the text you might want to keep. To allow rapid opening of the text file for replace mode editing, TextPro has three commands that will quickly insert editing room in the file.

These commands are... TAB - inserts and moves the cursor 5 blank spaces.

CTRL+TAB - inserts the same amount of blank spaces into the file leaving the cursor in it's original position.

These commands always insert regardless of editing mode, and can be used to open small areas in the file quite rapidly.

The third selection is the SHIFT+INSERT <open window> command, this will insert 255 spaces from the cursor position, each time it is pressed. This will help to do large text editing and entry in the middle of a large file.

TEXTPRO WORD PROCESSOR continued

Press this key as many times as needed to insert the needed space into the file. Do not use the INSERT MODE since this would defeat the object of opening the editing window in the first place.

Now we know how to insert text into the middle of a file, how do we delete the text or spaces which we have created. Firstly to delete the large window opening, use the SHIFT-DELETE (delete blank spaces) command and all the extra blank spaces between the cursor and the next word in the file will be deleted.

When in the replace mode the DELETE BACK-SPACE key replaces the character before the cursor with a blank space without moving the following text up behind it. In the inverse mode the text after the cursor follows the cursor back in the file.

There are three more commands for deleting text, the key press CTRL+SELECT-U is to delete text from the cursor to the top, as mentioned above when deleting a segmented file.

The next command is CTRL+SELECT-V this will delete text from the cursor to the end of the file, this is a quick way of deleting a large part of a file.

For deleting and moving text use the CTRL-D command. The command line will turn RED and the prompt "Delete (S,W,P,F): RETURN to exit" will appear in the message line. The "S,W,P,F" stands for: (S)entence, (W)ord, (P)aragraph, and to-(F)ind-string. This specifies the chunk of text in front of the cursor that you wish to delete.

Pressing these keys will delete the text up to the selected point and place it in the paste buffer. Any text previously stored in the in the paste buffer will be lost. You may choose text to delete until you press Return (or any other key than S,W,P, or F) to quit the delete mode and return to the normal editor. A (F)ind-string must be previously defined using the SELECT+CTRL-F(ind string) command, explained below.

Once you've returned to the normal editor and found that you would like to delete another paragraph or sentence but would like to keep the content of the paste buffer, you can use: the SELECT+CTRL-D(elete and append to buffer) command. The command line will turn green and the newly deleted text will then be added to the original contents of the paste buffer.

To replace the deleted text use the CTRL-R(eplace deleted text) command and the text in the buffer will be pasted back. You may paste the text back as often as you wish or move to another place in the document and paste it back. You may even load another file and paste the text into that file. The deleted text will remain in the paste buffer until you issue a CTRL-K(ill paste buffer) command which clears the buffer.

Now let me explain the Find and Change feature. TP has a very powerful search and replace feature that can search and replace ANY character or string up to 30 characters long. I know how good this is I used this feature on numerous occasions, and I also used the CTRL-G(lobal replace) command, to change over 100 characters in a very long text. These characters were a printer code in inverse which I entered wrongly, and using this global replace feature I could watch all the codes change to the correct code one by one, it was amazing.

TEXTPRO WORD PROCESSOR continued

To define a string to search for, press the SELECT+CTRL-F(ind string selection) command. Enter the string that you wish TP to search for and press Return. To find the selected string press the CTRL-F(ind string) command. There is no need to press Return, TP will immediately search out the first occurence of the string in the file.

To change a found string use the SELECT+CTRL-C(hange string selection) command. Just enter the string that you want to change the found string to, press Return, and you are ready to go. Each time the program finds the search string and you want to replace it press CTRL-C(hange found string) command and the found string will be changed.

Or at other times you may want to replace every occurence of one string with another without doing each one individually. Use the CTRL-G(lobal replace) command, just enter the string to find and the string to change to, pressing the Return after each selection. The search and replace operation will begin immediately and will proceed through the entire file.

There are still a few more commands which I would like to give a short description of. You can turn ON/OFF the CR in your text using the SELECT+CTRL-O(n/Off CR display) command. There are two more modes besides insert and replce modes, explained earlier, and these are "Text Mode and ATASCII Mode" toggle these editing modes with the ATARI key. The Text Mode is the Basic TP editing mode. When in ATASCII Mode, a number of commands are altered. The CTRL-W(here's the cursor) command is a print formatting aid in Text Mode. Any time this key is pressed the page, line number, and lines per page of the character under the cursor in the printed file will be displayed in the command line.

In the ATASCII mode CTRL-W(hat's the cursor) command gives the ATASCII value of the character under the cursor. For example, place the cursor on the CR and change into ATASCII Mode then press CTRL-W and the message line will display (Dec.#30 saves #155). Loading the .CNF file into the editor and changing to ATASCII mode, if you know what you are doing, you can change the screen colour or other decimal values. To change the decimal value press SHIFT-TAB and type in the decimal number.

TextPro doesn't have an inverse mode as such, in the ATASCII the CTRL-A (Switch) key will move to the left changing characters from normal to inverse and vice versa. In the Text Mode this key changes the case of alpha characters, inverse or otherwise.

Other editor commands are listed in the command summary in Section 12 of the manual. I wouldn't have been able to write about all these editor commands without the help of the manual.

In the next issue more editor commands.

ATARI CLASSICS
The Magazine for the Dedicated 8-Bit User

ENERGIZE YOUR 8-BIT: Plug into AC! ATARI CLASSICS, that is!
SUBSCRIBE NOW !!
Rates: Foreign (All) 3rd Class Mail $32 for 1 year
Europe/Mediterranean AIRMAIL $38 for 1 year
Payment by I.O.M or VISA - MASTERCARD
(Credit Card orders $2 processing fee under $100)
ATARI CLASSICS, 179 Sproul Road/RT.352, Frazer, PA 19355-1958
or Phone 313-973-8825.

# Letter Section

Dear Editors,

I have just put down your Jul/Aug 1993 Issue #4 newsletter, and all I can say is that it isn't only wine that improves with age, its also your newsletter, and disk. The expertise and knowledge by each of the authors, displayed on each page, makes me realize how little I know, and makes me want to learn more about our venerable 8 BITs.

Your disk is chock full of interesting programs, specially the CITIZEN group of programs on side 2. I read the note on the back cover which explains that you were short of space, therefore the Disk Content Page was missing, however, my thought is that a complete description of the program should have been included, and one of the articles should have been left out. If a disk has a full side made of one or two programs, they deserve full descriptions of the hows and why's. I managed to get a good sense of the program by the trial and error method, and was able to print some very nice picture examples to show THE OL' HACKERS at our next meeting, but I really can not tell them exactly what other features the program has, that I haven't discovered. In all probability your next issue will have a full description, but by then I'll be working on some other program, and may fail to come back to this program.

I hope you accept my comments only as constructive criticism, and not intended to knock your efforts. as I have told you, I look forward to each excellent issue, and hungrily pour over both disk and magazine as fast as possible, in anticipation of all the goodies that you are bringing. It is because of the work and efforts of your TWAUG, and THE OL' HACKERS among others, that the 8 BIT continues to thrive, even thoug it was abandoned by its parent, the ATARI Corp.

Atarily,

Alex Pignato,
President, and Editor of the OL' HACKERS A.U.G.,Inc.

Reply by Max

*Thank you very much for your letter which is very much appreciated. I also accept your comments, or constructive criticism as you call them. I will not be offended, because I do feel bad about not publishing the instructions in the newsletter. They are actually on the disk, the extender is .NOI, if the author added .TXT or .DOC these extenders would have been easier recognised as text files.*

*A leaflet will be enclosed in the newsletter with the instructions, and I will, in an other part of this issue, give credit to the authors of all the printer drivers.*

*Thank you again for your comments Alex, and we at TWAUG are really pleased for the pleasure we are giving to all our readers.*

Dear Twaug... Letter Section!

What no letters?

I've been enjoying the newsletter since the outset. One thing I would like to see printed... if it's possible... are pin-out diagrams and lead-configurations: to make printer leads for the 850 etc etc....

I've been a long-term user of Textpro. Using the 'Games Machine', with NO upgrade I have found it difficult to alter the .CNF file with the more recent versions! Hints? (I can't alter the screen-colour, for instance!)

Yours faithfully,

Joe King

Reply by Max...

*Thank you for your letter we are very pleased that you are enjoying reading our newsletter. As for publishing pin-out diagrams, watch the next issue there could be something interesting in it.*

*To change the background colour of the TextPro editor press CTRL-B and to reverse the sequence press SELECT+CTRL-B. To change the text luminance press CTRL-T and to reverse the sequence press SELECT+CTRL-T. To save the setting to disk press SELECT+CTRL-S and type the .CNF filename, but if you have any difficulties in achieving a good result, write to TWAUG enclosing a SAE and more details.*

---

David received a letter from a friend from Leven Scotland, Bill Jackson. He has a problem with TextPro version 4.0+ which has mistified him. David of course sitting, head in hands thinking, pass this letter on to Max the TextPro trouble shooter.

Bill deleted a couple of line from the end of a text with CTRL-D, this delete command put the text into the paste buffer, from where you can paste it back anywhere into the file. Now what happened next is a mistery to me as well, Bill printed out this letter and at the end of the letter was the deleted line of text. He said that the line was not to be seen on the screen.

I have had ghost text re-appearing after deleting to the end of the file with SELECT+CTRL-V, on earlier versions of TP, and it even printed out some of that text, but never from the paste buffer. I also know that earlier versions had bugs in, has anybody else had this problem where it printed some text from the paste buffer, it would be very interresting to know, write to TWAUG and let us know, please.

# LETTER SECTION

Dear TWAUG,

I have recently received issues 3 and 4 of your newsletter in exchange for a program I have written for you and I must say I'm impressed. I hope you can go on and keep the right balance of articles, reviews etc as these two issues. Do you offer a back issues service where I can buy issues 1 and 2, either with or without disk?

I have heard you hope to publish a book soon as well as some new commercial software. I wish you luck and I hope that they sell well.

I have a problem that I hope one of your readers can help me with. Recently my trusty XF551 disk drive half died on me. I say half died because it will no longer read or format side 2 of a disk when used as a true double sided drive with either SuperDOS or MyDOS 4.5, side 1 still works 99% of the time, so I came to the conclusion that there is a problem with the top set of heads on the mechanism. I spoke with a few people who suggested the XF551 mechanism is just an industry standard 5.25 360k one. So I bought one of them and replaced the faulty one but the drive wouldn't work at all. The busy light didn't come on and the head wouldn't step. Worst of all the company I bought the new mechanism from wouldn't take it back as they say it worked fine. So I am now stuck with a half working XF551 and a new 5.25 360k mechanism that I can't use (until I can get a Black Box and Floppy Board). My question is, can anyone help me solve this problem, either by getting the old mech to work or knowing a fix to allow me to use the new mech, or I would even consider buying another mechanism so long as it is an exact replacement for the Mitsumi mechanism used in the XF551. Can anybody help me? Please!

Finally I would just like to get a quick plug in for some of my software and hardware products that are available now or will be very soon (hopefully):-

Yorky 256k plug-in memory upgrade for the 800XL, gives you a 256 XE compatable computer with no soldering necessary, £50 + p&p, limited stocks.

Print-LAB, Graphics screen 8 designer with printer support for 1029, 1020 and Epson compatables available from Derek Fern, a specially modified and upgraded 128k version will be available soon.

Alien Blast, a NEW lightgun game that I am working on, still in it's early stages and will hopefully be available in advance of November's AMS show in Stafford (hopefully well in advance but you never know!).

Also I have written several PD programs and put together digi demo disks etc. Some titles to watch out for are Devil's Domain, Yorky Support Disk, The Simpsons demo no.1, with a no.2 in the pipeline.

Once again, keep up the good work and long may you prosper.

Yours Atarily, R. Gore.

Reply by Max,

*Yes a back issue service is available with or without the disk, just write to TWAUG with your request of the issue numbers wanted.*

*If there is someone who can help Richard please contact him his address is:*

*Richard Gore.*
*79 Sprotbrough Rd.*
*Sprotbrough*
*Doncaster*
*S. Yorks.*
*DN5 8BW.*

*TEL. (0302) 784642.*

---

Dear Those to Blame,

I have received issue 4 of your newsletter and I reckon that this is the best yet. Having read Max's letter section prompted me to write to you.

Firstly, it is great that you are going to write a new programming reference book. I have often tried to find Mapping the Atari without succes, so you can count me in as one of your buyers.

Secondly, a suggestion for a column for the newsletter. How about lowdowns on hardware accessories such as a modem, the Black Box, a Hard Drive. The articles could contain details like their availability, cost, different uses, accompanying software etc... I would find this very useful as I am at the stage where I would like to expand my system and put it to much more use in all sorts of fields.

By the way I thought that Wordweave is a brilliant concept and hopefully will be putting it to some use. Can you make stand alone stories and other applications with this instead of having to have the original program and the data on a seperate disk?

Yours Atarily

Bryan Zillwood

Reply by Max

*Well Bryan we are one step ahead of you, an article on the Black Box is in this issue, and there will be more write ups on hardware in following issues.*
*We also have the advert in the newsletter where the BB and other accessories can be obtained from plus prices*

# LETTER SECTION

## PAGE EDITOR PROBLEM

We received a letter from one of our subscribers, Ray Pawson, to let us know that there were a couple of problems with one of the programs on the Page Editor Set. They are both in the program called DOCREAD.BAS on disk two. Here is what he writes:

The first problem is that on running DOCREAD.BAS and pressing 'A' to print out the first document, an error code 170 at line 140 appears. This indicates, on checking the basic program, that the program (line 120) is looking for the file PAGEDIT.DOC whereas the file on the disk is PAGEDIT1.DOC.

The second error occurs on printing out the second file, PAGEDIT2.DOC; the print-out stops in the middle of page 5 when checking with a word processor shows the document to be 6 pages long. This for me was a little difficult to resolve, not being a programmer and being as thick as two short planks to boot, but I managed to fix it in the end by increasing the value of SZ in line 130 to 20000.

Reply by D.Ewens,

*Thank you for pointing these errors out to us, I have corrected the master copy in our PD library and suggest that anyone who has bought this program does the same.*

*I must admit that I did notice these errors in the DOCREAD.BAS file myself when I first got this program but I used my word processor and DD3 to layout and print the docs and clean forgot about correcting the other program. I think I am the thick one, not you.*

---

# MARK'S GAMES COLUMN...

## STATIONFALL by Mark Stinson

### Another Great Infocom

Anyone who knows me, or who may have read any of my articles and reviews will have undoubtedly guessed my bias towards adventure and particularly my preference of the Infocom range

I make no apologies for my leaning towards Infocom which, sadly, no longer produce adventures for the classic. If you are even remotely interested in adventure and have never tried an Infocom then you must buy one now, while there are still some available. I have stocked up on all those available from the various suppliers, and still have 12 to play. Why not contact Derek Fern of Micro Discount, Miles Better Software, or Page 6 and try one out? If you can obtain an Infocom you would be hard pressed to find a better adventure than Stationfall which is quite simply brilliant.

Stationfall is the sequel to the hugely successful Planetfall (although you need not have played this to play Stationfall).

## STATIONFALL continued

Along with the disk you will find the now famed selection of goodies contained in each Infocom.

There is the ever helpful official technical manual, which is a must for all players, especially the beginner. Along with this there is a supply of forms which will be needed during the game, a full set of station blueprints, and a badge sporting the caption 'boldly going where angels fear to tread'. With badge sewn firmly on to your tunic, and administrative forms in hand, you commence your adventure.....

Boldly Going

Once you have the disk booted up you find yourself on the SPS Duffy, a huge Federation Starship among whose crew you are but a lieutenant 1st Class (which is a step up from your rank as Ensign 7th Class in Planetfall).

Reading through your paper work you find a robot authorisation form, spaceship activation form, and an assignment completion form. You will see that this latter form is an instruction to fly to Space Station Gamma Delta Gamma 777-G 59/59 to collect '24 pallets of request for stellar patrol issue regulation black form binders request form forms' - thrilling stuff. If only life were that simple.

Before you depart on your mission you must visit the robot pool to appoint a companion on your mission. There are three robots to choose from, but however hard you try you always end up with the co-star on this epic mission, Floyd. Floyd is a boistrous little robot who was also your companion in Planetfall. He really is the central attraction in Stationfall, and hours of fun are to be had watching his comic behaviour. He is playful, moody, and at times unreliable on your mission, which makes him all the more 'human'.

The Departure

The reunion over, you are now ready to commence your mission to the space station and to get there safely you must fly a spaceship presently docked on the east side of the SPS Duffy. The truck is mostly automated so that you need not worry about elementary spaceship physics. You will, however, be required to enter the time code on the truck's keyboard. If you make an error here you will certainly regret it! I remember the reverse thrusters bursting into life after a short trip and the truck coming to a halt in empty space. You have no fuel, and your only relief is by way of an emergency beacon, operated by pressing the appropriate button. Having pressed this button I listened, with a cold shudder, to a recorded voice as it tried to calm me: 'there is nothing to worry about, nothing can go wrong (skip) go wrong (skip) .....'.

However, if you are successful in programing the spaceship accurately, you will shortly be greeted with the incredible sight of a huge Space Station looming up before you. You can see that attached to this miracle of technology is a seedy looking space village which has grown in an uncontrolled

## STATION FALL continued

manner over the years (obviously no Town Planners in this neck of the woods!). Before long the ship glides gracefully into a docking bay. Having collected your belongings, such as food and forms, you are ready to begin the quest.

You leave the ship with Floyd in tow, and enter the Space Station.

The Space Station

The Space Station is pretty big, so there's plenty to explore. One of the first things that strike you as you roam the place is the distinct lack of human activity. This is particularly strange when you study the blueprints and find that the ship's complement is usually 43, comprising 36 crew, 6 officers and a Station Commander.

The station is split into three main areas; the Central Command Module, which is on 9 levels; a Scientific Sub Module; and also the village. At various locations in the station you will find bits and pieces to aid you on your quest. For example, film spools are dotted about and these can be very useful sources of information when read on the machine in the library. The Commander's log also makes interesting listening, and gives you a further indication that something must be awry.

It appears from a recent entry that the crew were tracking an alien ship as it approached the Space Station. The ship, it transpires, was brought into one of the docking bays for examination. The remains of a long dead alien lifeform, and a curious pyramid were found aboard the ship. Extensive biological/scientific investigations ensued (as a second priority to the main raison d'etre of the Station - printing!).

Since the arrival of this ship and its strange cargo all of the electronic devices on board have been acting strangely. This is particularly concerning when you consider that all of the doors on the ship are electronically operated and soon start 'snapping at your heels'. The Space Station is also maintained by welder robots who constantly check the integrity of the hull. When they start to mistake you for a hole in need of repair don't hang about! Later, even Floyd seems to change.

You will very quickly realise that your original mission is no longer of primary importance, and that you have to now find the cause of these strange happenings. Should you fail, then life as we know it may never be the same again.

In Summary

Stationfall is an excellent adventure which will keep you on your toes, and which poses some real good problems to solve. Having said that, it is not so hard as to cause undue difficulty to the average adventurer.

If you can get a copy of Stationfall then I would strongly suggest you buy it immediately. Alternatively I am looking for a suitable Infocom to exchange for my original version (I would particularly like Zork II) so if you fancy a swap why not get in touch with TWAUG.

THE FULL SOLUTION

If you have not yet played Stationfall, then I would recommend that you skip this section as it will reveal all of the actions necessary to complete the game. If, however, you are stuck then you will find your answer(s) here.

E, N, INSERT ROBOT FORM IN SLOT, TYPE 3, S, E, OPEN HATCH, ENTER TRUCK, TAKE KIT, CLOSE HATCH, SIT, INSERT ACTIVATION FORM IN SLOT, TIME, ENTER CODE FROM SHEETS SUPPLIED, WAIT UNTIL YOU ARRIVE AT THE STATION, STAND, OPEN HATCH, LEAVE TRUCK, E, D, D, OPEN CAN, GET FORM, NW, GET DRILL, SE, U, U, U, U, NW, OPEN PRESSER, PLACE CRUMPLED FORM IN PRESSER, CLOSE PRESSER, START PRESSER, STOP PRESSER, OPEN PRESSER, GET FORM, E, D, D, SE, SE, E, OPEN KIT, OPEN BOTTLE, EAT SOUP, LOOK UNDER BED, GET STAMP, STAMP VILLAGE FORM, DROP STAMP, W, NW, S, INSERT VILLAGE FORM IN SLOT, S, S, NE, NE, SE, INSERT ID IN MACHINE, START MACHINE, TYPE 7, STOP MACHINE, GET CARD, (NEVER CARRY THE ID WHILE WEARING OR CARRYING THE BOOTS - THE MAGNETIC FIELD WILL RUIN IT), NW, SW, SW, N, N, NW, D, SE, INSERT ID CARD IN READER, N, GET GUN, S, W, U, SE, S, W, WAIT UNTIL FLOYD ARRIVES, FLOYD GET MEDIUM BIT, GET MEDIUM BIT, E, N, E, S, E, DRILL SAFE, REMOVE SMALL BIT, DROP SMALL BIT, INSERT MEDIUM BIT, DRILL SAFE, DROP DRILL, W, NW, S, S, S, NE, NE, NW, EXAMINE CEILING, OPEN PANEL, GET STICK, SE, DOWN, SW, S, SHOOT BOX, GET COIN, N, SW, GET CAN, NE, NW, N, N, N, SE, E, GET IN BED, WAIT, GET OFF BED, GET GUN, CAN, KIT, STICK, W, NW, S, S, S, SE, SE, NW, NE, U, N, E, BREAK MIRROR, GET FOIL, W, S, NW, OPEN CAGE, NE, SPRAY CAN, W, SPRAY CAN, W, SPRAY CAN, W, NW, NE, INSERT COIN IN MACHINE, TYPE 6, PLACE STICK OVER HOLE, GET TIMER, SW, SE, SPRAY CAN, NW, SPRAY CAN, SW, SPRAY CAN, U, SPRAY CAN, U, SPRAY CAN, SW, OPEN PULPIT, PRESS SWITCH, SPRAY CAN, HOLD LEASH, OPEN STAR, GET DIODE, D, DROP CAN, E, U, S, GET DETONATOR, OPEN IT, REMOVE BLACKENED DIODE, EAT GRAY GOO, DROP BLACKENED DIODE, NW, D, WAIT, WAIT, WAIT, FLOYD HELP ME (AFTER BEING SHOT), GET ALL, E, E, E, GET LAMP, W, S, SE, E, SPIN WHEEL, U, OPEN LOCKER, GET SUIT, WEAR IT, D, W, W SE, D, GET BOOTS, WEAR BOOTS, W, NE, OPEN INNER, ENTER INNER, CLOSE INNER, OPEN OUTER, TURN ON LIGHT, OUT, READ LABEL, PLACE CYLINDER IN BOTTLE, CLOSE BOTTLE, CLOSE KIT, IN, TURN OFF LAMP, CLOSE OUTER, REMOVE SUIT AND BOOTS, DROP THEM, OPEN INNER, IN, U, N, N, W, W, W, NW, SW, GET ALL, SE, SE, E, ATTACH TIMER TO DETONATOR, DROP THEM, OPEN KIT, OPEN BOTTLE, GET CYLINDER, ATTACH IT TO DETONATOR, PLACE CYLINDER IN HOLE, LOOK, SET TIMER TO 25, W, WAIT, E, GET KEY, W, NW, NE, NW, N, N, N, U, (LIGHTS GO OUT), TURN ON LAP, GET BOARD, D, S, S, W, GET JAMMER, ATTACH BOARD TO JAMMER, E, S, SW, (PLEASE NOTE, YOUR INVENTORY MUST BE: FOIL, JAMMER, 20 PRONG FROMITZ, KEY, LAMP, GUN, KIT, DROP KIT, U, U, U, U, UNLOCK BIN WITH KEY, OPEN BIN, GET ALL, OPEN GRATING, ENTER AIR SHAFT, SET JAMMER TO 710, TURN IT ON, D, D, D, D, D, D, KICK GRATING, TURN OFF JAMMER, U, SHOOT FLOYD, PUT FOIL OVER PYRAMID.

Simple really!

# The BLACK BOX

from Computer Software Services

The Black Box is a add-on board for the Atari 800XL, and 130XE 8-bit computers. It is a T-shaped board that plugs into the PBI port of the XL computer, or the ECI and cartridge ports of the 130XE. Connectors for both types of computers are built-in to the Black Box, so no adaptor boards are necessary. A cartridge port is available on the board itself for 130XE users, since the board plugs in where cartridges are normally added. The board is 12 inches wide and 3 inches deep, sitting back 3 inches from your computer. It has two types of switches, two push-buttons, and two toggle switches on the top.

The Black Box provides many useful functions. The three primary functions are: RS-232 serial modem port, Parallel printer port, and a SASI/SCSI hard disk port. You can now purchase the BB with the floppy board ready fitted, this will add a floppy disk port for connecting 3.5" or 5.25" floppy drives.

The RS-232 port provides the full RS232 specification signal levels for a modem, or other serial device. It emulates the Atari 850 interface very closely, but goes beyond by providing 19,200 baud capability. The R: driver is built-in to the Black Box, so it does not use ANY user memory!

The Parallel Printer port interfaces to most all Centronics-type printers. You may assign the printer number and linefeed options from within the Black Box's configuration menu. The Black Box also provides you with a printer buffer, if the board or your computer has extra memory. A printer buffer allows you to quickly dump your file to be printed into the buffer memory, then go about your business as the Black Box sends the data to your printer; a real time saver! The Black Box will use either its own RAM (if you order the 64K version), or the 130XE extended memory banks; it's all controlled by the configuration menu.

The Hard Disk port is the real reason for the design of the Black Box. You may connect most any hard disk controller that is SASI or SCSI compatible, or drives with embedded SCSI controllers. It is totally compatible with the current versions of MYDOS and SpartaDOS (which both have a limit of 16 megabytes per logical drive), but a newer version of MYDOS should be out soon that is capable of 48 megs per drive. Combine that with nine drives, and that's over 400 megs available at one time! The Black Box also provides a conversion toggle for drives capable of 512 byte sectors only. Many of the embedded drives have this limitation, and previously were unusable. The Black Box splits each 512 byte sector into two 256 byte sectors, so your DOS will still only see what it requires. Another advantage is storage space. Many drives/controllers will give you more storage when using 512 byte sectors, some as much as 15% more! The following drives/controllers are known to work with the BB: Adaptec 4000A and 4070 (SCSI), Xebec 51410 and 51410A (SASI), Western Digital 1002SHD (SASI), OMTI 3520 and 3527 (SCSI), SMS 5001 (SASI or SCSI) and Seagate embedded SCSI drives.

A partition is defined as a part of the hard disk which is seen by the computer as a separate disk disk drive. Since many hard disks are very large, it is useful to create several partitions of the drive, instead of one single drive, as your DOS sees it. The Black Box goes one step further in not only letting you define the partition for each of your 9 available drives, but allows you to have a list of up to 96 partitions, with names! Since a partition can be very small, you can make up several small partitions of 720 sectors (the same length as a standard floppy disk), and sector-copy any of your non-protected programs to these partitions. Now you can swap that partition in as drive 1, and boot your program at hard disk speed!

The configuration menu is the 'heart' of the Black Box. You can enter the menu from anywhere you are by simply pressing one of the buttons on the board. You may now edit the hard disk configuration, exchange drive numbers, enable/disable the modem and printer ports, or go into the 6502 monitor. After you are finished, pressing ESCAPE will put you right back into the program you were using! No memory or screen display is destroyed by using the menu!

The 6502 monitor is very handy for machine language programmers. How often have you wondered where your program was, or what caused an apparent 'lock-up'? Entering the monitor will show you all the processor registers, and display the disassembly of the instruction it was about to execute when you entered pressed the button. Users of MAC/65's DDT will feel right at home with the monitor's use.

The Black Box has other 'goodies' in it. Any communication with your floppy drive will be in high speed if you are using a XF-551, a modified 1050, or a happy 810. This will work with just about ANY DOS or utility!

A text or graphics printer dump of your current screen may be done at any time by pressing one of the buttons on the Black Box. (The graphics dump is only available for dot-matrix printers capable of graphics.)

You may write-protect ALL of your hard disks by flipping another switch on the board. This can be a real life-saver when running a new piece of software. The Black Box provides disk I/O tones with separate pitches for disk reads and writes to your hard disk, so you can hear what's going on! This option may be disabled within the configuration menu.

The build in Task Master sector editor is a powerful editor. It contains a sector copier featuring multiple copies, automatic formatting and uses all available memory. Since the Black Box provides Ultra Speed data transfer to modified floppy drives, this makes for fast disk duplication. It can handle up to 16MB hard disk partitions, (even in the sector copy mode.) Since the Black Box gives you the capability to do screen dumps to your printer, you can make hardcopies of your editing.

## COMPUTER EXPERIENCE

by Bill Jackson,

I have been meaning to contribute an article to TWAUG since I first received issue No.1, but as I have very little programming skills, I wasn't at all sure what kind of article it would be.

While reading through issue #4 of TWAUG I was pleasantly surprised to see the article entitled 'WHY AM I DOWN ON IBMs' by John Kasupski. I really enjoyed this article, and it got me thinking, why not just write an article about my experiences with computers in general.

So, here goes:-

It all started in about 1985 when my brother in law bought a Commodore 64 and I played a game similar to O'Rily's mine. I was hooked, I wanted one of these things. The problem was that at the time (and many times since) I was unemployed. This meant that I couldn't afford one. Then in December 1985 Atari had a large sales push offering Atari 800XL'S at VERY good prices. I managed to persuade the wife to let me buy an 800 XL with cassette deck (I knew absolutely nothing whatsoever about an Atari but I read an article in some comsumer mag and it got a very good review, so that was good enough for me). Those first few weeks were sheer hell. The Atari manual was pathetic. I spent most of my time tearing my hair out, wrestling with this and swearing at that, then I discovered the 'Atari User', a godsend.

The Atari User was truly a life saver, it didn't solve all of my problems but at least I wasn't ALONE.

I used my 800XL mainly for playing games and I also spent many hours typing in game listings from Atari User. I tried to learn BASIC but as soon as I encountered anything mathematical that was me beat.

Around about 1987 my computing world was about to take on a new twist. The local minister ran a community program (remember them) part of which was a computer dept. This computer department had 800XL'S with DISK drives amongst other things, so to cut a long story short, I volunteered for the job of 'teaching computer literacy', sounds grand doesn't it. All it meant was that I had to learn various software packages (on my own, again) from the manuals and then show some computer novices how to use said packages and the computer.

A big deal I said, could I use an Atari ST ? I was asked, Why, yes I lied.
I had read about the wonderful Atari ST in the Atari User and indeed Page 6 but I had never actually clapped eyes on one.

Here we go again I thought, out with the manual and get stuck in. It's just as well that I took up 'home?' computing as a hobby, I couldn't imagine me putting in hundreds of hours overtime for free on any real job!!

I learned enough about the Atari St to get me by fairly quickly, again the Atari User and indeed Page 6 were life savers in this respect. The ST'S were far removed from the 800XL.

Part of my job was doing a couple of hours, two nights a week looking after the 'Computer Club'. This was a nightmare! it involved refereeing a dozen or so kids fighting over half a dozen various types of computers. If memory serves me right the run-down was something like this:- one Atari ST, three Atari 800XL'S, one Commodore 64 (with disk drive), two BBC thingys, and an old Apple II (I avoided this like the plague, as I didn't know how to use it and it didn't have a joystick). Of course the computer department had more than one Atari ST, but we wouldn't let the kids loose on these, heaven forbid.

Fortunately I didn't have to learn too much about the Commodore and the BBC's, just enough to boot games really and a little (very) BASIC programming on the BBC. The old Apple II was virtually obsolete, it was originally used to run 'Wordstar' a wordprocessor I would do anything to avoid.

In those early days of the ST there were a few problems worth a mention, these were the days before built-in disk drives, and just to confuse us poor part-time idiots, the powers to be, bought some single sided disk drives and some double sided disk drives. The result was chaos, people would format disks in a double sided drive and someone else would try and run said disk in a single sided drive, and not surprisingly a few mishaps would ensue. Data disks were getting corrupt all over the place, pandemonium. We eventually abandoned the single sided drives and achieved some sort of sanity.

Another source of problms was printers, we had three well used printers all different makes, this led to much messing about with dip switches and fiddling with printer drivers and a mountain of wasted fanfold lying in the corner, Ah, those were the days.

We setled down to some serious work learing packages such as First Word (and later First Word Plus), Degas (and later Degas Elite) a couple of early spreadsheets and databases ( I can't recall their titles) which were pretty hopeless and were dropped like a stone.

Degas Elite was far and away the most popular package both with the staff and the students, particularly when using an STFM (which we had acquired) and a colour telly, this gem of a drawing prog kept everybody quiet for hours, great!

By far the most interesting programs starting to emerge for the Atari ST was the Desk Top Publishing programs. The department bought a completely new system especially for D.T.P. this consisted of an Atari1040 computer and a Panasonic P6 24 pin printer.

COMPUTER EXPERIENCE continued

The first software package they bought was entitled 'Publishing Partner', I was just getting into D.T.P. when my years C.P. job came to a close (Aha..) so I waved goodbye to the ST's and went home to my humble 800XL.

Eventually I managed to get my hands on a new 800XL with 1050 Disk Drive this saved my sanity and put new life into my hobby.

The next couple of years passed by with me bashing away at my 800XL keyboard, delving into utilities, games and even the odd adventure and writing to pen-pals.

In 1989 I decided to return to the 'Agency' (as it was now known) as part of yet another government scheme, work all week full-time for an extra tenner over and above dole money (I must have been mad). This time I was going as a student to polish-up my computer skills and catch up on the ST world.

I didn't recognise the place when I went back, the boss sat in a nice little office and had a 3 meg Atari ST with a hard drive and a laser printer, changed days indeed. The main work area was chock-full of Atari STs and various printers (including my old pal the Panasonic P6). I spent the next six months learning various new (to me anyway) programs such as Superbase, Fleetstreet and my favorite D.T.P. prog, Timeworks.

Just off the main work area was a little office with a nice looking computer sitting on a desk, what's this then Steve, I asked, a pain in the arse he replied. I was intrigued. It was some kind of Amstrad with a hard drive and a colour monitor (I can't remember the model No.) an I.B.M. compatible computer which I had heard a lot about but never encountered. I thought it might be handy to learn something about this thing and also M.S.Dos. Fool!

I'll start off with something familiar I thought. I spied an old friend lurking in the hard drive's directory, First Word Plus no less. I proceeded to boot-up F.W.P. to say the least I was very dissapointed with the Amstrad, it was very slow and lacking response so I gave up on F.W.P. and thought I'd try a go at typing in a nice little BASIC program that I found in the printers manual, that would print out a nice colour hardcopy.

Now, as I have mentioned earlier I'm no good at programming, but I have typed in a few hundred progs in my time. It took me quite some time to get this 'little' prog up and running but eventually I managed it. As you may have gathered, I wasn't impressed with this machines BASIC either. I dare say that if someone was to start off using this machine and had never used an ST then they would get used to it, but after being spoilt on the ST's it was a big let-down.

I saved the best for last, M.S. Dos. I spent quite some time wading through the manual and mucking about with floppys and the hard drive, (note:- I couldn't do any damage as the machine was intended for training and all files on the hard disk were available on floppy's) I'm glad I did as it was valuable experience, but I'm glad I don't have to use it for a living. So much for my I.B.M. compatible experience, it made me glad to have a 'humbie' 8-bit Atari.

Well then, you might ask what's all this leading to? Well I dunno really. What with my fairly extensive knowledge of the Atari STs, and my obvious admiration for them, why don't I have one? Simple, I don't really NEED one. Sure I'd like one, but I couldn't afford the system that I would want. I wouldn't be happy with just a standard Atari 520, I'd want the moon. Indeed I'm happy with my Atari 130XE and U.S. Doubler 1050's and even my 7-pin 1029 printer. They're streets ahead of Commodore 64s, BBCs, Apple IIs, Amstrads and many more expensive systems.

Well there you have it, my entire (nearly) computer history, I hope I havn't sent anyone off to sleep, bye.

## HINTS and TIPS.

Well, here I am again with what I hope are some more useful bits and pieces of information that I've picked up when browsing through some of my old magazines.

Although these are tips that I have found useful to me in the past, I would much rather be trying out tips that some of you out there had sent in to be included in this section. Why not get your thinking caps on and see what you can come up with. I'm sure if you really thought about it, you could come up with enough info to keep this section going for at least another 6 issues. Well, here goes with this issue's selection.

WHICH DOS?

Sometimes it would be helpful to know which DOS you booted with. With one or two PEEKS from BASIC you can easily find out. Use this table to determine the DOS version.

| PEEK() | VALUE RETURNES | DOS VERSION |
|---|---|---|
| 3889 | 0 | SpartaDOS 2.3e |
| | 13 | DOS 4.0 |
| | 15 | SpartaDOS 1.1 |
| | 19 | Atari DOS 2.5 |
| | 76 | Ataro DOS 3.0 |
| | 78 | Atari DOS 3.0 |
| | 89 | SpartaDOS 3.2d |
| | 108 | MYDOS 4.0 |
| | 207 | OSS OS/A+4.00 |
| | 221 | MYDOS 4.50 |
| | 238 | Atari DOS 2.0S |
| | 238 | OSS DOS XL 2.3 |
| | 244 | Atari DOS XE |
| 1804 | 0 | OSS DOS XL 2.3 |
| | 124 | Atari DOS 2.0S |

Just type PRINT PEEK(3889) and press RETURN at the READY prompt in BASIC, and compare the number printed on your

# HINTS & TIPS continued

screen with this chart. Note that Atari DOS 3.0 may return either a value of 76 or 78 at location 3889. Both Atari DOS 2.0S and OSS DOS XL 2.3 show a value of 238 at location 3889. To distinguish between the two, check location 1804.

## INSTALLING DOS.

To put DOS on a non-DOS disk from Basic, simply open a file called DOS.SYS for output, and then close it. The DOS you booted with will be written to the disk, and you can now boot with that disk. You won't be able to get a DOS menu from it -- that part of the operating system is contained in the file DUP.SYS which must be copied seperately.

The following commands to install DOS on drive 1 work with Atari DOS 2.0, DOS 2.5, MYDOS and quite possibly with other operating systems too:

```
100 OPEN #1,8,0,"D1:DOS.SYS"
110 CLOSE #1
```

## FILE READER

Here's a routine that will allow you to read disk files without having to TRAP the end-on-File error.

```
100 DIM A$(1000):CHAN=3
110 OPEN #CHAN,4,0,"D:FILENAME.TXT"
120 INPUT #CHAN,A$:PRINT A$
130 IF PEEK(1535+CHAN*16)<>3 THEN 120
140 CLOSE #CHAN:END
```

This routine should work with all text files except one opened for a directory. The variable A$ should be DIMmed big enough to hold any amount of text in your file between carriage returns. CHAN is set equal to whatever channel you wish to use.

That's all for this time, hopefully I'll be including some of your hints and tips next time.

A way of mimicking Turbo basics DO-LOOP loop in standard Atari basic is....

```
FOR I=0 TO 1 STEP 0:NEXT I
```

Variable I can never equal 1 due to the zero step therefore the loop will never end. It is then said to be an infinite loop. The loop can of course be exited by using the POP command.

This is also a neat way of preventing a program from ending. It can be added to the end of an existing line rather than the usual method of taking up a whole line as for example....

```
240 GOTO 240
```

This is important in a five liner where you may not have a spare line.

A way of mimicking Turbo basics DO-LOOP loop in standard Atari basic is....

```
FOR I=0 TO 1 STEP 0:NEXT I
```

Variable I can never equal 1 due to the zero step therefore the loop will never end. It is then said to be an infinite loop. The loop can of course be exited by using the POP command.

This is also a neat way of preventing a program from ending. It can be added to the end of an existing line rather than the usual method of taking up a whole line as for example....

```
240 GOTO 240
```

This is important in a five liner where you may not have a spare line.

# FIVE LINER

## 256 COLOURS by John Foskett

This is a small demo program that displays all 256 colours of the Atari pallet on the screen at once.

Its display is basically a mode 9 screen with 16 columns of differing brilliance in a single colour with the darkest column on the left. The first column is colour 0, the next colour 1, the next colour 2, etc. up to colour 15 on the extreme right. A DLI is used to change the background colour 15 times down the screen and a VBI is used to reset the DLIs controlling register. This method allows the actual colours used to be included as basic data which may be altered to achieve different effects.

LISTING BREAKDOWN

Line 10 sets mode and loads data into page 6.
Lines 10 & 20 sets up display list.
Line 20 Sets up VBI and DLI vectors.
Line 30 Draws mode 9 screen columns.
Line 40 VBI and DLI data.
Line 50 Colour data.

```
10 GRAPHICS 9:FOR I=0 TO 45:READ J:POKE
1536+I,J:NEXT I:D=PEEK(560)+256*PEEK(56
1):FOR I=16 TO 186 STEP 12:IF I=100 THE
N I=I+2
20 POKE D+I,PEEK(D+I)+128:NEXT I:POKE 5
4286,0:POKE 548,0:POKE 549,6:POKE 512,1
1:POKE 513,6:POKE 54286,192
30 FOR I=0 TO 79:COLOR INT(I/5):PLOT I,
0:DRAWTO I,191:NEXT I:FOR I=0 TO 1 STEP
0:NEXT I
40 DATA 72,169,0,133,77,133,203,104,76,
138,194,72,138,72,234,234,234,166,203,1
89,31,6,141,26,208,230,203,104,170,104,
64
50 DATA 16,32,48,64,80,96,112,128,144,1
60,176,192,208,224,240
```

# FIVE LINERS continued

MULTI-FUNCTION VBI by John Foskett

A small VBI routine written to provide several important features for adding to any basic program. The routine includes the following features....

DISABLES THE BREAK KEY

Ensures the break key is permanently disabled. Modes may be changed without having to re-disable it.

DISABLES THE ATTRACT MODE

Ensures the attract mode is permanently disabled.

DISABLES THE CONTROL-1 TOGGLE

Ensures the control-1 stop-start toggle is permanently disabled.

KEYBOARD UPPER CASE LOCK

This feature provides a way of ensuring that only upper case characters can be entered from the keyboard. Location 203 controls this....
POKE 203,0
    Keyboard lock enabled.
POKE 203,(non zero)
    Normal use of the keyboard.

FLASHING CURSOR

A flashing cursor is provided together with flashing inverse video characters. Location 204 controls this....
POKE 204,0
    Normal steady state.
POKE 204,(non zero)
    Flashing enabled. Different values give various flash rates and patterns. An even flash pattern is provided when values are multiples of two, 16 gives a good flash rate.

LISTING BREAKDOWN

Line 10 is a REM statement.
Line 20 Temporarily disables the break key and loads the data into page 6.
Line 30 Sets VBI vector and the control registers.
Lines 40 & 50 contain all the data.

```
10 REM Multi Function V.II
20 GRAPHICS 0:POKE 16,64:POKE 53774,64:
RESTORE 40:FOR I=0 TO 59:READ J:POKE 15
36+I,J:NEXT I
30 POKE 54286,0:POKE 548,0:POKE 549,6:P
OKE 54286,96:POKE 203,0:POKE 204
,16
40 DATA 72,169,64,133,16,141,14,210,169
,0,133,77,141,255,2,165,203,208,8,141,1
82,2,169,64,141,198,2,230,209,165
50 DATA 204,240,17,165,209,37,204,197,2
04,240,9,208,12,141,243,2,104,76,138,19
4,169,2,76,43,6,169,0,76,43,6
```

MARK's 5-LINER

When Dave Ewens asked me if I could write a 5-liner for this issue I said yes, sure, without really having any idea what I could do which was useful and only took up five lines of code.

Well, I'm afraid I am not much of a programmer really, so I have had to resort to machine code to do something useful in five lines! Don't get me wrong, I'm not much of a machine code programmer, but I am trying hard to get to grips with it so this seemed the perfect opportunity to have a go.

The program listed below will, hopefully be of use to all you games programmers out there as it is a character set mover. The program first reads a short machine code routine into page 6 (starting at address 1536). It then moves the top of user RAM to page 151 and the machine code copies the character set from ROM to RAM page 152 (the extra page is needed to keep the set clear of some of the Atari's internal functions). Finally, location 756 is POKEd with 152 to point the computer to page 152 for the new set.

Some points to note are:

1. If you use GR.7 or GR.8 alter the POKE 106,151 to POKE 106,144 (giving a lower RAMTOP).
2. Any GRAPHICS command will alter location 756 so each must be followed with a POKE 756,152
3. If, at any point in your program you alter location 106 (to lower RAMTOP when using GR.7 or GR.8 for example) always issue a GRAPHICS command (whichever you are working in) otherwise some strange results will occur.
4. I would leave the line numbers as listed and issue a GOSUB at the start of your program as this will speed your execution time up.

All you need to do now is alter whichever characters you wish to define by poking the relevant 8 bytes with a new value (for a good article on doing this see Page 6 issue 55). As the program below is written in Machine Code you should find it is extremely fast (try it as is!).

```
32000 RESTORE 32007:FOR A=0 TO 77:READ P:POKE 1536+A,P:
NEXT A
32010 DATA 169,0,133,203,169,224,133,204,169,0,133,205,
169,152,133,206,160,0,162,4,136,192,255,208,1,202,132,
207
32020 DATA 134,208,160,0,177,203,145,205,198,207,165,20
7,201,255,208,10,198,208,165,208,201,255,208,2,104,96,
230
32030 DATA 203,165,203,201,0,208,2,230,204,230,205,165,
205,201,0,208,2,230,206,192,255,208,210
32040 POKE 106,PEEK(106)-9:GRAPHICS 0:X=USR(1536):POKE
756,PEEK(106)+1
```

I would like to acknowledge Stephen Williams, whose series in Page 6 have helped my understanding of Machine Code, and whose MOVE RAM program forms the basis of the M/C routine. I would also like to acknowledge Ann O'Driscoll whose article in Page 6, issue 55, provided the inspiration for the above program.

# VERTICAL BLANK INTERRUPT

by NIR DAREY

Let me start explaining what Vertical Blank is. You probably know that the time it takes the television to draw a full picture on a screen is 1/50 of a second (in the U.S.A it's 1/60 of a second). The TV keeps drawing the picture again and again even if it changes. The routine that draws the picture on a screen is already written in the ATARI O.S. Since the computer is running at 1.79 Mhz, there is spare time from when the computer finishes drawing the picture until it's starts again, this spare time we can use to insert our own routine. Any graphic changes being made during VBLANK will be shown smoothly, but the most important thing is that you can run a VBI routine at the same time as your main program!. (For example, play music, do smooth scrolling etc...) There is also a Horizontal Blank which occurs after the TV has finished drawing one line and goes to the next line. There are two stages of VBLANK, the first is called "IMMEDIATE" which is always executed, and the second called "DEFERRED". There are two ways to set a VBI to your own routine.

The first way is to set the VBLANK registers to the start address of your own routine. 546,547 ; Immediate VBLANK registers 548,549; Deferred VBLANK registers.

The second way is to load the Y register (LDY) with the low byte address of your routine, and the X register (LDX) with the high byte, then load Accumulator (LDA) with 6 for Immediate VBI, or 7 for Deferred VBI and do JSR 58460. I recommend the second way to set your VBI routine because if you change the VBI registers and a VBI occurs the computer will crash!!!. Another thing you should always remember is always tell the computer at the end of your VBI routine, to go back and do the O.S. VBI routine:
JMP 58463 for Immediate VBI.
JMP 58466 for Deferred VBI.

Now let's take a look at a small example the following source is written in MAC65:

```
10      *=1536
20      PLA
30      LDY #<VBLANK
40      LDX #>VBLANK
50      LDA #07       ;deferred VBI
70      JSR 58460
80      RTS
90 VBLANK LDA 20
100     STA 53274
220     JMP 5846
```

Here is the same routine for the BASIC user:

```
10 ADDR=1536
20 READ DAT:IF DAT=-1 THEN 50
30 POKE ADDR,DAT:ADDR=ADDR+1
40 GOTO 20
50 X=USR(1536):END
60 DATA 104,160,11,162,6,169,7,32,92
70 DATA 228,96,165,20,141,26,208,76,98
80 DATA 228,-1
```

Some technical notes:

Immediate VBI updates the, Internal Realtime Clock register (RTCLOCK 18-20),
Attract Mode register (ATTRACT 77), DRKMSK 78,
Changing Color register (COLRSH 79),
System Timer One registers (CDTMV1 536,537),

Then the Time Critical register (CRITIC 66) is checked, if it's set the O.S. returns from the interrupt. If it hasn't set then, the O.S. goes to stage two of the VBI.

Deferred VBI updates the:
Horizontal Position of the Light Pen register (PENH 54284),
Vertical Position of the Light Pen register (PENV 54285),
Display List address registers (DLISTL,DLISTH 54274,54275),
Priority register (PRIOR 53275),
Colour of Players and Missile registers (COLPM0-3 53266-53269)
Playfield Colour registers (COLPF0-3 53270-53273),
Colour of the Background (COLBK 53274),
Character Set address register (CHBASE 54281),
CHACTL register (54273),
TRIG0-3 registers (53264-53269),
POT0-7 registers (53760-53767),
PORT A register (54016),

# Another FIVE Liner

## ROM/RAM EXAMINER by John Foskett

The following "5 liner" was one of the first programs I ever typed in on my Atari 8-bit. I seem to think that it was originally published in Antic Magazine, but unfortunately I cannot be sure. I also do not know who the author was.

The program itself allows you to examine in ASCII format, the contents of the ROM and RAM. Simply run the program from BASIC, and when prompted, type in a start address and follow the prompts.

```
10 GRAPHICS 0:DIM H$(1),K$(1):? "Enter
start address":INPUT A$:? "How many lo
cations to examine":INPUT Z:Z=(Z/8)
20 FOR C=0 TO 2:? :? A;" :";:FOR A=A T
O A+7:P=PEEK(A):H$=CHR$(P):IF ASC(H$)(
32 OR ASC(H$))125 THEN H$=" "
30 ? " ";H$;:NEXT A:NEXT C:IF C)Z THEN
? :? "Press S to stop, C to change spe
cs, RETURN to continue":INPUT K$
40 IF K$()"S" AND K$()"C" THEN GOTO 20
:IF K$=S THEN END
50 RUN
```

# TEN LINER

## DISK COMPANION by John Foskett

A mini-program which provides four important disk I/O funtions without any prompts or warnings. Use with caution. The functions are as follows....

(1) Format single
(2) Format enhanced
(3) Unlock files
(4) Delete files

(1) and (2) are self explanatary

(3) unlocks all files on the disk

(4) Prompts for a file name to delete, wild cards "*" and "?" may be used for multiple deletions.

```
1 GRAPHICS 0:POKE 16,64:POKE 53774,64:
TRAP 10:DIM F$(12),FN$(14),L$(36):FN$=
"D:":L$="━":L$(36)="━":L$(2)=L$:POKE 7
52,1
2 ? "K))↑↑↓DISK COMPANION↓":? L$:? "1.
 FORMAT SINGLE)3. UNLOCK FILES":? "2.
FORMAT ENHANCED)4. DELETE FILES":? L$
3 POKE 764,255:CLOSE #1:OPEN #1,4,0,"K
:":GET #1,K:CLOSE #1:K=K-48:IF K(1 OR
K)4 THEN 3
4 ? "↓OPTION ";K;" SELECTED↓":IF K=1 T
HEN ? "FORMATTING DISK IN SINGLE DENSI
TY":XIO 253,#1,0,0,"D:"
5 IF K=2 THEN ? "FORMATTING DISK IN EN
HANCED DENSITY":XIO 254,#1,0,0,"D:"
6 IF K=3 THEN ? "UNLOCKING ALL FILES":
XIO 36,#1,0,0,"D:*.*"
7 IF K=4 THEN POKE 752,0:? "ENTER FILE
 NAME TO DELETE":? "↓D:↓------------↑↑
++++++++++";:INPUT #16;F$:GOSUB 9
8 ? "↓PRESS ANY KEY TO EXIT":POKE 764,
255:OPEN #1,4,0,"K:":GET #1,K:RUN
9 POKE 752,1:? :FN$(3)=F$:? "↓DELETING
 '";FN$;"'":XIO 33,#1,0,0,FN$:RETURN
10 CLOSE #1:POKE 752,1:? "↓↓↓ERROR ";P
EEK(195);" ENCOUNTERED":GOTO 8
```

# BUY & SALE SECTION

## BOOKS FOR SALE.

Atari DOS 2.5 XF551 disk drive owner's manual.

Atari 130XE owner's manual

Atari DOS XE owner's manual

The three above books £10 each plus £1.50 P&P.

The Atari 130XE handbook by Lupton + Robinson, £5 plus £1.50 P&P.

The following American Handbooks.

Inside Atari Basic by Bill Carris. 164 pages.

Atari Basic XL edition by Albrecht, Finkel, Brown. 386 pages

£10 each plus £1.50 P&P.

Contact: James Cutler
44 Water Street
Great Harwood
Lancashire
BB6 7QR
or phone:0254-887311

## WANTED.

Does anyone have a 1027 printer power unit for sale? Willing to pay reasonable price.
Write to: Mr. P. Hampshire.
12 Treelands Walk
Ordsall Park
Salford
Manchester M5 3FX.

## CONTACTS.

Lately, I have been having some problems with our local postal service and I seem to have lost touch with many of my contacts. If any of my regular contacts have written to me during the last couple of months and have been waiting for a reply from me, would they mind sending me a letter and let me know.

Contact: Mr. Ray Robertson.
19 Littlebury Green
Basildon
Essex SS13 1PA

# D.I.Y. AND THE 1050 DISK DRIVE

By John Foskett

## WHILST PROGRAMMING

During programming one evening in April 1992, saving my work to disk at intervals of about 15 minutes or so as one should, I discovered to my horror that on one occasion error 144 was the result. I attempted to save the latest version of my work several times which all resulted in error 144. I inserted a different work disk into the drive and again attempted the save and again the same error resulted, error 144, at each attempt. I inserted yet another of my work disks and again attempted the save only to be greeted yet again with error 144. Having realised that saving my program was impossible, I wrote down on paper all that I could remember that I had done to the program in memory since my last successful save and switched my equipment off so that I could investigate the problem further.

Since all disks failed at the same time, it cannot be due to individual disks, such as finding bad sectors. Therefore, it must be something to do with my 1050 disk drive itself.

## THE DRIVE

My first task was to remove the cover which is achieved by inverting the drive and removing the six screws securing it. The cover is then removed by turning the drive back the right way up and carefully lifting the cover from the back, pulling it forward, lifting it clear of the drive.

Once opened, the first step is to inspect the tiny pressure pad which is positioned above the read/write head upon a pivoted, spring loaded arm. This tiny pad is the usual reason why a disk drive fails to read from or write to a disk.

With a disk inserted in the drive and the drive closed, it can be seen that the pressure pad makes contact with the actual disk itself. There must be a small gap between the surface of the disk and the tiny cup that supports the pressure pad. This is to ensure that only the pressure pad touches the disk and not the cup itself, the gap should be about 1mm.

## MY PROBLEM

Upon inspection, I could see that virtually no gap at all exists between the pressure pad cup and the surface of the floppy disk. This is therefore the most likely cause of my problems. But even if it is not the main cause of the problem, it must still be corrected. I could also see that this could be the cause of another annoying problem that has begun to plague me, that of a horrible squeal which is generated as the disk rotates, obviously generated by the cup actually touching the disk.

Over a period of time, during use, the pressure pad gets compacted and one method of reviving it, is to unfold a paper clip and very carefully roughen up the surface of the pad with it.

This process however, proved to be a failure in my case, since the pressure pad had literally worn itself away. The only remedy was replacement, but with what and how ?

A CASSETTE

After an hour or more in very deep thought, it occurred to me that perhaps another item that uses a pressure pad could be put to a good (or better) use. I realised that every cassette uses a pressure pad and that I have many old "C15" computer tapes that I haven't used since becoming a disk user several years ago. Here then, was the answer to my problem, make a new pressure pad.

## HOW TO PROCEED

The first step then, is to remove the spring loaded arm that supports the pressure pad. This is achieved by very carefully pushing out the small spindle that the arm pivots on taking extreme care not to touch the read/write head immediately below it. The spindle can be easily removed by using a pointed tool (a 2" panel pin can be used). The spindle is best pushed out from the main drive motor side of the disk drive and finally withdrawn from over the stepper motor using a pair of long nosed pliers. Great care must be taken to prevent the tiny spring from flying out of the drive under its own tension whilst withdrawing the spindle. This could be done by using a short length of cotton (taken from the wifes sewing box) tied to the spring before the spindle is withdrawn.

With the pressure pad arm removed, it merely remains to remove the tiny pressure pad support cup from the arm itself. This can be seen to be a small seperate component carrying the actual pressure pad. Removal of the pressure pad cup is achieved by gently squeezing together the sides of what looks like a large screwdriver slot and allowing the cup to fall free.

Once the tiny pressure pad cup is removed from the arm, the pressure pad itself (or what remains of it) can be easily seen. Another attempt to revive the pressure pad using a pointed tool should be tried before replacement is finally considered.

It is important that all traces of the old pressure pad and its adhesive are removed prior to replacement. This is straight forward since it can easily be seen that the original pressure pad is secured using self adhesion.

## MAKING A NEW PRESSURE PAD

The next step is to obtain an old cassette (or buy one, the price must be worth it). The cassette must then be very carefully opened and its little pressure pad removed. This is normally a pressure pad secured to a tiny thin phosphor bronze spring strip.

The next procedure is to seperate the pressure pad from its spring clip which is achieved using a sharp knife or a scalpel blade. The actual method of removal will, of course, depend upon the design and make of the cassette (I used a Boots "C15" cassette).

DIY 1050 continued

Once the pressure pad is removed, the remainder of the cassette can then be discarded. The pressure pad, once removed, can be seen to be about 6mm x 4mm x 2mm thick.

Using a pair of scissors or a sharp knife, the next step is to cut the pressure pad into a square of about 4mm. It is accurate enough to do this by eye. The four corners are removed next, again by eye to form an octagonal shape which is near enough circular to fit into the pressure pad cup. The new pressure pad should be slightly larger than the internal dimension of the cup to provide a secure fixing.

The next step is to cut a piece of double sided, self-adhesive tape about the size of a postage stamp (any make will do, I used Scotch). The protective backing is removed from one side and the pressure pad firmly placed into the centre. It is important that the side of the pressure pad that was originally in contact with the tape within the cassette should eventually make contact with the disk. The double sided tape is now trimmed around the pressure pad to provided a circular pressure pad with a self-adhesive backing.

Ensuring that the cup is free from any remnants of the original pressure pad, remove the protective backing from the new pressure pad and firmly press it into the cup. The surface of the new pressure pad should be about 1mm above the level of the top of the cup.

It now only remains to re-assemble the components back into the drive. It is a simple matter to re-insert the cup complete with its new pressure pad back into the arm. But it is far from easy to re-assemble the arm, spindle and the spring. However, with care and patience this can be successfully achieved.

For this reason, it is a good idea to sketch on paper before removal, the exact position and orientation of the spring, noting the amount by which it unwinds upon removal, which is normally half a turn. There are three positions in the lower moulding for the spring which provides three different settings for the pressure of the pressure pad, this should also be noted.

## IT WORKS

Having completed the replacement of the pressure pad inside my 1050 disk drive, the next step is, of course, to test it. Before replacing the top lid, I re-connected the drive and switched my equipment on. I booted the computer using DOS 2.5 and attempted to format some blank disks, which I found would format perfectly, without any errors. I could see the pressure pad as it rode over the surface of the rotating disk looking much healthier than before.

## THE DRIVE IN USE

I have used my 1050 disk drive for well over a year now since replacement of the pressure pad. By observation, I can see that the new pressure pad is still looking very healthy and working well.

## MY WORK DISKS

Listing the directories of my work disks that I originally attempted to save my program to, proved impossible. But I could list the directories of other disks. I wondered if they could have become corrupted in some way when attempting the saves. By using a disk sector editor, I found that they were well beyond repair, I had no choice but to reformat them and to start again. Fortunately there was nothing else of great importance on the disks therefore a reformat was not too unbearable.

## TAKE A TIP

Always use at least two work disks and swap them often which could avoid loosing a lot of work if errors should occur during a SAVE. If errors do occur while SAVEing, never risk corrupting all work disks by attempting further SAVEs to them, always use a blank disk or one of lesser importance if further attempts are to be made.

---

# WHAT NO DISK CONTENT...!

*As you will have noticed there was no disk content in the last issue of the newsletter, and I apologise for it. I was pushed for time as I was away for about four weeks.*

*The printer drivers documentation should have been printed out in the newsletter. This is the reason for my apology and also for not crediting the authors for their work.*

*The ARCed file on the issue disk contains: 2 printer drivers for Panasonic-24 to use with AtariWriter+, one by Richard Gore and the second driver by Ralph Bradley. The documentation for both drivers is included in the ARCed file, the doc file by R.Gore has the extender .DOC and is easily recognisable, but the other file by R.Bradley has .AWP as the extender, and not every body would recognise this as a doc file.*

*The Citizen Swift-9 Colour printer drivers, by Ralph Bradley, has its document also on the disk. I didn't notice the docs at first, as their extender doesn't tell you at a glance that it is a text file, especially when you're not a Mini Office user. The extender is WO2, if it had been .TXT or .DOC, I probably wouldn't have had any complains.*

*Again, SORRY folks for my cock up.*

# CRACKING THE CODE

by Keith Mayhew and Roy Smith

Re-printed by M. Gerum

This article first appeared in "The UK ATARI Computer Owners Club" later renamed "MONITOR"

Part 5

We are now to the point where we can start the more practical side of machine code programming, using an assembler and running some simple examples. It is recommended that you experiment with such simple programs. Do not be dishartened if things do not go as expected. You will learn a lot from your mistakes, such as save a copy of your program in case it crashes!

## WHY AN ASSEMBLER?

We already know that each instruction has been given a standard three letter mnemonic to help remember its purpose. In the last issue there was a complete list of these mnemonics and their associated op-codes. The term 'hand assembly' is given to the process of manually converting an assembly language program into machine code by the use of such a table. However, the task is made more tedius by the fact that most mnemonics have more than one op-code entry in the table, each allowing for the different addressing modes the instruction can be used in. Then there is the added complication of changing all numbers into one base, say decimal, for entry into the computer. There are many more reasons why hand assembly becomes difficult and slow, that's apart from any mistakes you can easly make when working with pages of numbers.

Assemblers overcome all the problems associated with hand assembly because the process of converting assembly language (of mnemonics, labels, etc.) into machine code becomes just one command. Assemblers have many more advantages than just converting the assembly language that make the development of a progrm a lot easier. Most assemblers are supplied with a powerful editor to create and modify assembly language programs and a 'de-bug' or 'monitor' program that allows you to look at memory locations, dis-assemble machine code back into mnemonics and many more useful features that will help you to test and find bugs in your programs. Even when writing small subroutines you will find that an assembler is a luxury over doing things by hand. I would advise you to invest in one of the several assemblers available if you intend to get to grips with the code, in the long run it will save you time and frustration!

This series will be centred around the ATARI Assembler Editor cartridge which is perhaps the most popular among users, it is not the best, but it is easy to use and quite comprehensive. Apart from the actual speed of assembly, other assemblers tend only to differ in minor respects, or offer features which only an advanced programmer may find useful, such as macro's. An excellent compromise over a commercial assembler is USERCOMP, although a little limited in that most of the features associated with an assembler are missing, you can enter all the mnemonics and is great for writing small routines or for experimenting with.

## GET THOSE FIELDS IN FILE

There are two types of files which are associated with an assembler, the source file and the object file, these reside on either disk or cassette. The source file represents the assembly language program which you create and modify with an editor. This is analogous to how you enter and save a BASIC program except that the program is usually saved as pure text, ie. exactly as you typed it.

When you decide you want the machine code for the assembly language you command the assembler to compile the source file. The source file will be read and an obect file created, this will be actual machine code, on a disk system you could 'binary load' this file (directly from DOS) into memory. Remember that any changes you want to make to your program should be made to the source file, re-compiling will then give the updated version of the machine code, in the object file, ready for execution. At this point the fundamental difference between a language like BASIC and an assemble should be clear: BASIC is an interpretive language and executes statements direct from a source file, line by line, and therefore BASIC has to be resident to run the program. An assembler is a compiling language and works with two files - source and object - and converts all the statements in the source file into directly executable machine code in the object file, therefore the assembler does NOT have to be resident to run the object code.

We will now look at the structure of the source file in some detail. The file consists of lines of text, each line is thought to consist of fields of characters. Here a field consists of one or more adjacent characters, each field is separated by one or more spaces - analogous to the words in a sentence. The first field is normally a line number, this has NO effect on the assembler what-so-ever.

Line numbers are used by most editors to keep track of lines and as such are completely ignored by an assembler if they are present. We shall, therefore, call the first field either the one at the start of the line or the one following the line number (by one space), depending on which type of editor you are using. A line can have from one to four distinct fields.

## THE LABEL FIELD

A label must start with a letter and can have any mixture of letters and digits following it, forming a 'name' for the label - similar to a variable name in BASIC, except there is usually a limit of up to six characters only. Labels are used to represent numeric values; once a value has been assigned to a label then where ever the label is used its value will be substituted during the assembly of the program. The first field on a line is reserved for a label name - you will see the use of labels soon.

## THE OPERATION FIELD

The second field is the operation field and is where the familar op-code mnemonics are written, such as LDA. Psudo-operations, also known as assembler directives, are also written in this field and are used to control the operation of the assembler when compiling your source code.

# CRACKING THE CODE continued

## THE OPERAND FIELD

Operands are written in the third field and are the data for what is in the operation field. The type of operation will determine if an operand is necessary or not. Any label which was previously defined (by placing its name in the first field) can be used in the operational field - the effect is the same as writing the actual value of the label in place of it, so labels are effectively substitutes for fixed values, or constants.

## THE COMMENT FIELD

Comments are simply a string of characters commenting the operation of the program, similar to the REM statement in BASIC. The start of a comment is usually signified by a semicolon (;), when this is reached the rest of the line is completely ignored. The comment field is an exception in that it can start at any place on a line. They can take up a whole line or can be placed at the end of any existing line. The use of comments helps describe what a program is doing and should be used quite liberally throughout any program as documentation. If you find an old program at the bottom of a draw then the comments will prove invaluable to remembering what it did and how!

## NEAT FIELDS

At least one space must be used to skip a field, if the tab key is used to space fields out you will not have to worry about the exact spacing and as a bonus you will find that they line up neatly from line to line. The following shows some possible field layouts.

```
100 LABEL LDA #$3A;Minimum spacing
200       PLA     ;Operation only
300 PLA           ;Same but with
                   minimum spacing
```

From now on tabs will be used to space these fields out. Listing 1 shows a complete program. The program loads the accumulator with the value zero, stores this at location 3000 hex and then returns to the calling program. The third line down contains:- '*=  $0600', the '*=' is an assembler directive which sets the origin or location counter of the assembler to 600 hex. This causes the machine code to be generated from location 600 hex onwards. Such a command must precede any program to determine where the code will be stored and can be used anywhere else, thus splitting the code up into different sections of memory if desired.

Listing 2 is the output from the assembler during the assembly, it is an exact copy of Listing 1 except that two extra columns of numbers have been appended to the left hand side. These show the contents of the location counter and the contents of the memory, respectively. From this we can see that 'A9' the op-code for 'LDA' is stored at location 600 hex and that '00', the operand, or data, is stored in the next location of 601 hex, the next line shows the count or address at the next location of 602 hex as expected. Therefore, by looking at the second column of numbers we know the exact contents of the object file, ie. the actual machine code for our program and the address of any particular byte.

## LABEL IT

Listing 1 does not have anything in the first field, ie. labels. If a label is placed in the first field then it is assigned a value. This value will be the contents of the location counter, ie. the current address. Another method exists for assigning a value to a label by using the '=' or equate directive in the second field, the operand following this will then be the value assigned to the label. It should be noted that the assignment of a value to a label can only be made ONCE during a program. Once given a value, the label can then be used in the operand field as the data to whatever is in the previous field. When assembled the value of the label is substituted for the labels name.

Labels are used to give names to memory locations or data. One advantage of using labels is that they aid your memory, instead of remembering or looking up a value repeatedly, a meaningful name can be given to the value and then the label name can be used where ever needed. Another advantage is obvious, if you want to change a value then you haven't got to search the entire program changing the value many times, instead you would only need to change the value of the label (at the start of the listing).

Listing 3 shows the use of labels to represent memory locations and data. The program adds an increment to a memory location and stores it back again. Line 120 assigns the value 4000 hex to SUM, this is the address where the number to be incremented is stored. The next line assigns the value of 13 hex to INCR, this represents the amount to be added to the sum. The rest of the program is as normal except that where an address or data is needed the label name is used instead. When assembled, the label names are simply substituted for their values in the operand field. Therefore the effect is exactly the same as if, for examle, line 160 read: 'LDA  $4000'.

If we now wanted to change the address of SUM we only have to change it on line 120, as opposed to having to change both line 160 & 180, if we had written the actual value of $4000 there. Listing 4 uses two labels, the first is defined in the same way as before and is the address of a count. Line 140 has the second label, ADD, however this time it preceeds a 6502 mnemonic. In this case the value assigned to ADD is the address of the instruction it preceeds, which happens to be the start of the program, ie. 600 hex.

Line 180 is a jump instruction, it jumps to the address held in ADD, so in this case it is the same as writing: 'JMP $600'. Thus the program continues in a never ending loop adding two to COUNT every time it goes around the loop. Labels can, of course, be placed anywhere in a program, a jump or branch can then be made to the label, instead of trying to work out which location the instruction is stored in. The use of labels in this way can be thought of as 'marking points' in a program which can then be jumped to the label or 'marker'.

## ODD EXPRESSIONS

Yet another advantage of an assembler is to evaluate expressions in the operand field. For instance; LABEL =2*3+1 will evaluate the expression on the right hand side to be seven and then assign the value seven to LABEL.

# CRACKING THE CODE continued

Such an expression can contain other label names which are already defined, these expressions can also be used as an operand to a 6502 instruction. This facility does not provide a short cut to doing multiplication and division in machine code; expressions are evaluated once at assembly time and are then used in programs as a fixed value - just as a label has a fixed value. The use of these expressions does, however, save you having to use a calculator to work out a constant, instead the computer works it out for you at assembly time. A useful expression is the single quote followed by an ASCII character, when the expression is evaluated the ASCII value of the character becomes the operand - this saves looking up characters in tables of ASCII values and writing the value down. Listing 5 demonstrates the use of this. Assuming the location of CHAR contains the ASCII value of a digit between 0 and 9, the program will place the actual number that character represents in location NUM. The ASCII value of 0 is 48 decimal, the characters 1 to 9 have the consecutive values of 49 to 57, so to convert the character to the number it represents we simply have to subtract the value of ASCII 0, from it. Line 170 does the subtraction, the 0 is evaluated to be 48, the ASCII for the character 0, the result is then stored in location NUM. For instance, if the ASCII character was 9 then 57 would be in location CHAR, by subtracting 48 from this we get the actual value of 9, stored in location NUM.

## GETTING IT TOGETHER

Having seen a lot of examples on the features of an assembler, we will end by writing a program which, when called by BASIC will multiply two integer numbers together and return the answer to BASIC. Although not being spectacular, it will hopefully bring together a lot of ideas and principles covered so far in this series and should be fully understood.

## USING THE USER

Before we can start we need to look at the mechanism by which BASIC allows the user to call machine code routines - by the aptly named 'USR' command. This command also allows numbers known as parameters, to be passed back and forth between the two. The general form of the command is;

X=USR(ADDR,PAR1,PAR2...)

Where 'ADDR' is the address in memory where the routine starts and 'PAR1', 'PAR2', etc. are the parameters to be passed; the parameters are separated by commas and the number of these can vary from none to as many as BASIC will let you type on one line. All the numbers should be integers between 0 and 65535 (although BASIC does round-off any none-integers). An integer, in the same range can also be passed back into the variable X of the USR command.

The numbers BASIC passes are saved on the stack. By doing this we dont have to worry about exactly where they are stored in memory, to retrive these numbers in the machine code we use the PLA command which takes the last entry of the stack and puts it into the accumulator. The first number pulled back tells us how many parameters BASIC stored on the stack, eg. if zero, then it tells us that there were no parameters supplied by the user.

This can be used to see if the actual number of parameters is equal to the number you expected, if not, the error can be delt with. Most of the time it is ignored and assumed that the user knows how many parameters he should supply, however using this method means that if you do give the wrong number of parameters it will almost definitely crash on you, so be careful. Once the parameter count has been taken off the stack, the parameters are then removed, two bytes for each. The first PLA will give the high byte of the first parameter in the USR command, the second PLA will return the low part of this same parameter, the rest are taken off in the same manner until all are removed. If everything went well, then the return address will be left on top of the stack ready for an RTS to take you back into BASIC. To return a value to the BASIC variable used in the USR command then, you simply store the low part in the location 212 and the high part in 213 (decimal). On return, BASIC sets the variable equal to the value of this two byte integer.

## DOING THE MULTIPLICATION

The easy way to multiply two numbers together is simply to add one of them repeatedly to a result, the number of times will be determined by the other number, for instance: 2*3 is the same as 2+2+2, ie. 2 added on three times. This is obviously very inefficent, especially for large numbers, as a lot of addition will be involved, better methods do exist but we will stick to this one for its simplicity. Using our method, we need to add the multipland to the result by the number of times indicated by the multiplier. For the example of 2*3, the multiplier is 3, thus we add 2 onto the result three times to get the answer of 6. However, this is exactly the same as 3*2, now the multiplicand is 3 and the multiplier is 2, this means that we do one less addition than before ie. 3+3. This is better illustrated by 1*255 which takes 255 additions of 1 instead of 1 addition of 255 for 255*1. The method, or algorithm, for our multiplication now becomes: Firstly, if the multiplier is greater than the multiplicand then swap them over, we then zero the result and add the multiplicand to the result by the number of times as the value of the multiplier.

Listing 6 shows the completed assembly program, the following discribes how you might go about entering it into an assembler to compile it. If you have an assembler editor cartridge then you would type in everything except the two columns on the far left, the comments are, of course, optional. If you have a different assembler then you would need to change a few minor things, see the appropriate manual for some help. Table one gives some guidelines for conversion between most of the popular assemblers. Before assembling the code you would save a copy of this as a source file (disk owners shoud note that the extension of .SRC is often used to denote the source file), assembly of the program is then done to an object file (for disk the extension of .OBJ is widely used). Assuming there were no errors then the contents of the object file would be the same as the second column of Listing 6 and would load into addresses shown in the first column, ie. page6 (600 hex) onwards.

# CRACKING THE CODE continued

## HOW IT WORKS

The equates on lines 120 and 130 set up two labels, MLTPND will hold the multiplicand and is at location CB hex, RESULT is set to point to location 212, this is where BASIC picks up the value for the variable in USR statement. Next the location counter is set so that the code produced is placed from 600 hex onwards. The first PLA gets the number of BASIC's parameters into the accumulator, however, this is discarded, as is the high byte of the first parameter, this leaves us with the lower half in the accumulator after the third PLA and is saved as the multiplicand by the following STA instruction at location MLTPND, ie.CB hex. Two more PLA's leave the lower part of the second parameter in the accumulator and is saved in the X register by the TAX instruction as the multiplier. Only the lower part of each parameter is saved so that neither number can exceed 255, by doing this the result will not need more than two bytes which is the maximum size we can return to BASIC'S variable.

We now want to see if the multiplier is less than the multiplicand to minimise the number of additions. A copy of the multiplier is still in the accumulator at this point, the compare instruction then subtracts the multiplicand from it, leaving only the N,Z and C flags affected. We want to skip the swap over if the accumulator was less than the data, ie. the comparison left a borrow condition because the larger data was subtracted from the smaller value in the accumulator. If a borrow occured then the carry (C flag) would be cleared, the next instruction BCC, would then branch over the following instructions to NOSWAP, thus not doing the swap. You can see by looking at the machine code, on the left, that the assembler has converted the address of NOSWAP (611 hex) into the offset of 05, following the opcode of 90 for the BCC instruction, yet another thing the assembler does for you!

If the branch failed then the instructions on line 240, 250 & 260 swap the multiplier and the multiplicand over, the accumulator is loaded with the multiplicand and then the multiplier, in the X register is saved as the new multiplicand, finally the accumulator is moved back into the X register as the new multiplier. Which ever path was taken, when the LDA instruction is reached at NOSWAP, the multiplier will be in the X register and the multiplicand will be in MLTPND with X being the lesser of the two. At NOSWAP the accumulator is loaded with zero and is stored at location RESULT (212), as the result will take two bytes we must also zero the high part at the next location. Rather than define another label to point to location 213. RESULT+1 is used, this is equivalent to 212+1 which is location 213, the high part of the result.

The multiplier is in the X register so that it can be used as a count, every time an addition is done we will take one from the count until it reaches zero. The test for this count being equal to zero is made on line 300, which compares the X register to the number 0, if it is then the 'branch if equal' (BEQ) instruction will take us to EXIT which returns to BASIC. The test is made first in case the multiplier is zero, if so then the result will also be left as zero. Assuming the multiplier isn't zero then lines 320 to 350 clear the carry and add the multiplicand to the low byte of the result. If the carry is left clear then a branch is made to NOCARY, else one is added to the high byte of the result by the increment instruction on line 370.

This method of seeing if the carry is set and then incrementing the high byte by one is exactly the same as adding with carry. (ADC) zero to the high byte, however, this would require a load and store instruction, thus we save two bytes! All that happens then, is one is taken from the count by the decrement X instruction and a jump is made back to the comparison instruction at MLTPLY to continue the multiplication. The loop will continue like this until the count in the X register has been taken to zero, once this happens it will return to BASIC and the answer, held at RESULT and RESULT+1, will be placed into BASIC's variable.

## RUNNING THE PROGRAM

To run the program it will have to be in the locations starting at 600 hex (page 6). If you assembled your own copy of this program then you could load the object file into memory, for disk owners life is simple, just nip into DOS and binary load the object file, then jump back to BASIC. If you are using the assembler editor cartridge with cassette then beware of an error. You cannot CLOAD the object file as stated in the manual, instead you will need a short routine, such utility routines will be delt with in the next part of this series. If you can't load your object program or dont have an assembler yet, then Listing 7 is a BASIC program which reads the hex data and stores them in memory after they have been converted to decimal for the POKE statement, if you have made a mistake in the data a 'checksum error' will be printed out, if so, then re-check the DATA statements. Having saved the program, type RUN and the machine code in the data will be poked into memory, after a short delay the message 'Data in memory' will be displayed. You can now test it by typing:

```
ANSWER=USR(1536,10,24)
```

then typing:?ANSWER will give the result of 240. You can change the two parameters, but dont alter the first number of 1536, this is the decimal for 600 hex, which is the start of the program.

## UNTIL NEXT TIME...

In the next issue I will tie up some loose ends, including looking at an improved multiplication routine, and then start some new programs and topics. Have fun experimenting until then, and if you haven't already done so, rush out and buy that assembler you were promising yourself!

```
0100 ;Simple program showing the effect of
0110 ;using the tab key to space out fields.
0120         *=     $0600    ;Start code at 600 hex.
0130         LDA    #0       ;Zero location
0140         STA    $3000    ;3000 hex.
0150         RTS             ;Return to caller.
```

Listing 1

# CRACKING THE CODE continued

```
                0100 ;Simple program showing the effect of
                0110 ;using the tab key to space out fields.
0000            0120         *=     $0600   ;Start code at 600 hex.
0600 A900       0130         LDA    #0      ;Zero location
0602 8D0030     0140         STA    $3000   ;3000 hex.
0605 60         0150         RTS            ;Return to caller.
```

Listing 2

```
0100 ;Program to show the use of labels to
0110 ;represent memory locations and data.
0120 SUM    =       $4000   ;Address where sum is stored.
0130 INCR   =       $13     ;Increment value.
0140        *=      $0600   ;Set location counter.
0150        CLC             ;Clear carry for addition.
0160        LDA     SUM     ;Get current sum.
0170        ADC     #INCR   ;Add on the increment.
0180        STA     SUM     ;Store the result back.
0190        RTS             ;Return.
```

Listing 3

```
0100 ;Program showing the use of labels
0110 ;to 'mark points' in programs.
0120 COUNT  =       $4000   ;Holds current count.
0130        *=      $0600   ;Set location counter.
0140 ADD    CLC             ;Clear carry for addition.
0150        LDA     COUNT   ;Get count.
0160        ADC     #2      ;Add on increment.
0170        STA     COUNT   ;Store new value back.
0180        JMP     ADD     ;Jump back to start.
```

Listing 4

```
0100 ;Program to find the 'value' of
0110 ;an ASCII number.
0120 CHAR   =       $4000   ;Holds ASCII character.
0130 NUM    =       $4001   ;Holds value of the number.
0140        *=      $0600
0150        SEC             ;Set carry for subtraction.
0160        LDA     CHAR    ;Get ASCII character.
0170        SBC     #'0     ;Subtract value of ASCII 0.
0180        STA     NUM     ;Save as a number.
0190        RTS             ;Return.
```

Listing 5

```
                0100 ;Simple multiplication of two single byte
                0110 ;integers by repeated addition.
00CB            0120 MLTPND  =       $CB     ;Multiplicand.
00D4            0130 RESULT  =       212     ;Result for BASIC.
0000            0140         *=      $0600   ;Start on page 6.
0600 68         0150         PLA             ;Discard number of data.
0601 68         0160         PLA             ;Discard high byte.
0602 68         0170         PLA             ;Get low byte.
0603 85CB       0180         STA     MLTPND  ;Save as multiplicand.
0605 68         0190         PLA             ;Discard high byte.
0606 68         0200         PLA             ;Get low byte.
0607 AA         0210         TAX             ;Save as multiplier.
0608 C5CB       0220         CMP     MLTPND  ;Compare both.
060A 9005       0230         BCC     NOSHAP  ;OK if MLTPLR<MLTPND.
060C A5CB       0240         LDA     MLTPND  ;Swap MLTPLR
060E 86CB       0250         STX     MLTPND  ;for MLTPND
0610 AA         0260         TAX             ;so that MLTPLR>= TPND.
0611 A900       0270 NOSHAP  LDA     #0      ;Zero the result
0613 85D4       0280         STA     RESULT  ;low
0615 85D5       0290         STA     RESULT+1;and high.
0617 E000       0300 MLTPLY  CPX     #0      ;If MLTPLR=0 then
0619 F00F       0310         BEQ     EXIT    ;exit loop.
061B 18         0320         CLC             ;Clear carry.
061C A5D4       0330         LDA     RESULT  ;Add in MLTPND
061E 65CB       0340         ADC     MLTPND  ;to the result
0620 85D4       0350         STA     RESULT  ;and save back.
0622 9002       0360         BCC     NOCARY  ;No carry so branch.
0624 E6D5       0370         INC     RESULT+1;Else adjust high byte.
0626 CA         0380 NOCARY  DEX             ;Decrement count
0627 4C1706     0390         JMP     MLTPLY  ;and go back.
062A 60         0400 EXIT    RTS             ;Exit back to BASIC.
```

Listing 6

Listing 7

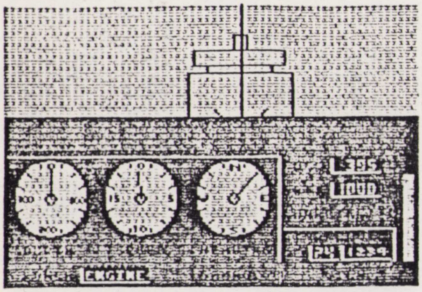
```
10 DIM HEX$(16)
20 LINE=10000:TRAP 100
30 READ HEX$,CHKSUM:SUM=0:J=0:START=
1536
40 FOR I=1 TO 15 STEP 2
50 D1=ASC(HEX$(I,I))-48:D2=ASC(HEX$(
I+1,I+1))-48
60 NUM=((D1-7*(D1>16))*16+(D2-7*(D2>
16)))
70 SUM=SUM+NUM:POKE START+J,NUM:J=J+
1:NEXT I
80 IF SUM=CHKSUM THEN LINE=LINE+10:G
OTO 30
90 PRINT "Checksum error on this lin
e:"
95 LIST LINE:END
100 PRINT "Data in memory."
10000 DATA 68686885CB6868AA,1026
10010 DATA C5CE9005A5CB86CB,1254
10020 DATA AAA90085D485D5E0,1254
10030 DATA 00F00F18A5D465CB,960
10040 DATA 85D49002E6D5CA4C,1212
10050 DATA 1706600000000000,125
```

| Description | Atari Assm/Edtr Cartridge | Synapse Syn Assm Disk | O.S.S. Mac 65/Bug 65 Disk or Cartridge | Atari Macro Assm Disk |
|---|---|---|---|---|
| Set origin (Location counter) | *= | .OR | * = | ORG |
| Value of location counter | * | * | * | * 0 |
| Equate (assignment) | = | .EQ | = | = or EQU |
| Define bytes/ characters in memory | .BYTE | .AT<br>.HS<br>.AS | .BYTE<br>.SBYTE<br>.DBYTE | DB<br>DC |
| Define words in memory | .WORD | .DA | .WORD | DW |
| Skip bytes | *=*+ | .BS | *=*+ | DS |

Table 1. Assembler Conversion Chart

# GATO

by Mark Fenwick



Gato from Atari corp is a World War II submarine simulator, supplied as a Rom for instant play. Another good quality Rom from Atari just before they decided to drop the 8-bit.

You have been assigned to a GATO-class submarine, these vessels being the Americans backbone of the sea's during World War II. Your mission is to patrol the South Pacific and destroy as many enemy vessels as possible before returning to the subtender ship for fuel and supplies. Gato comes complete with a an informative manual of instructions, which once read are easy to follow. A clear break down of all keys and their uses are put together on one page for easy reference.

On loading Gato the menu screen appears showing your submarine, underneath are your various options. "1" Will commense game play. "2" Gives a demo run through the game, giving an illustrated explanation on controls and the enemy. "3" Gives a breakdown on the history of the Gato submarines. It's a good idea to take a look at the demo to give an idea of game play.

On selecting the play game option you must enter the current date, (Well, any date will do really) remember month first! From here we go to the second menu screen where you can select difficulty, sound (for morse code messages), day or night, reset captains log, number of players, to which I was surprised to find you can have up to four players! long time since I've seen this. Once you've selected your required options you go on to the game.

This screen is split to 50/50(ish). The bottom half shows controls, depth, speed, heading etc and the top half shows your view over the ocean. From this screen you can check out the sonar as well as your position in the Pacific. You can easily identify yourself and subtender from the enemy as they are a different colour. When you first checkout your position on the chart you have the option of using a Rapid Sub Deployment, in other words you can move from one side of the map to the other via entering the desired password and coordinates. A little reminder of Star Raiders galactic chart!

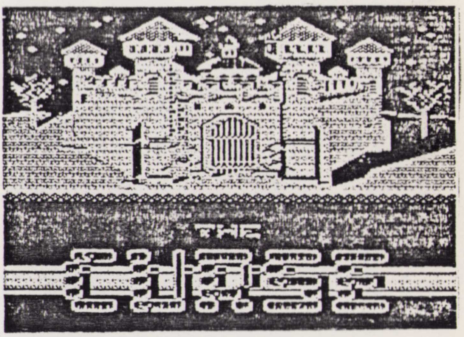
## GATO continued

The graphics are well portrayed in the sense of the 3D effect. Ships don't just look side ways on on the horizon, you can actually see the ship at a 45 degree angle. You can also view a ship at the side or behind you to give an all round view. As you complete each level you'll soon discover the degree of difficulty increases, in that ships are chasing you! Also in later levels missions are given to you in morse code only, a chart for translating is included, though thankfully.

Gato is a very good attempt at a simulator and after a few plays you soon get to grips with the controls. As simulators go it does have an element of atmosphere with sharp detailed graphics and average sound. How it compares to the likes of Submarine Commander and Silent Service, well, better in the first instance but Silent Service may just have the edge simulator wise. For playability Gato is better, though maybe just short on the odd touch. I would recommend this as a game to buy though my version cost only £7.00 as apposed to Micro Discounts price of £18.95! So it would be an idea to try other outlets such as Gralin International for £9.00 or B & C computervisions USA, 10 dollars + shipping (See Page 6 for addresses)

# THE CURSE

Review by Mark Fenwick




The Curse or Klatwa, (For the Polish out there!) from Avalon Poland, this has to be without doubt the best game to hit the U.K. since Op Blood.

As with all imported software the instructions are brief, a small inlay sheet is enclosed with the disk to get you to the basics. The main story to the game is that...Your country has been plagued with misery for seven years. The locals have turned to thieves stealing from their neighbours losing all sight of what is good or evil anymore. This curse was apparantly cast by the evil wizard, Ruler of Darkness. Your objective is to find and conquer the evil wizard so lifting the curse and returning your country to good as it once was. Hmm, is this Wizard John Major by any chance ?

The Curse is a grahpic adventure, hang on who said "Oh no, not another N,S,E and W job!" No, quite the opposite, you'll see none of that sort of thing here. I for one am not an Adventurer so it was a welcomed relief when I first played the Curse, everything is controlled via the joystick with the

## THE CURSE continued

On booting the disk you'll soon see the castle you are about to embark upon, below this is the title followed by credits to the authors, accompanied by a great sound track. Pressing fire or start will open the castles main gate taking you on your adventure.

The screen takes up two thirds play area while the bottom third shows relative information. The first screen shows a room with your player standing in the middle. The graphics here are quite breath taking given the fact that we're on an 8-bit not an St as the programmers may have you think. Attention to the detail of your player as well as the 3D effect background could not possibly have been done any better. This effect gives the room great depth, and at the back of this first room is a door which once entered reveals another room. Here we see various items such as a table and chair, cupboard, windows, pillars not to mention a tiled floor. On the screen you'll also notice an arrow which you must use to carry out all tasks, ie. Go, Look, Open etc Simply point the arrow at the table and a window menu will drop down on the lower part of the screen. Here you can choose what you wish to do from what is listed to select, just move your arrow and press fire. Any part of a screen can be looked at more closely. As you start off you see a door in the background, point the arrow at the door and press fire. A window menu drops down to ask if you wish to Open or Cancel, press for Open and our hero will casually stroll to the door and open it. The second room holds a table with a drawer which contains a spell book and cap, these items can be picked up for later use. The spell book is needed to open certain doors and carry out other tasks. Each time you look closely at an item it will be blown up to give a clearer view.

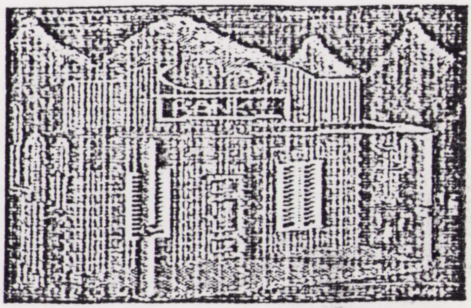
The Curse is an excellent graphic adventure, the fact that you see your character walk around achieving his tasks makes it that more appealing to the non adventure player. I find most adventures a bit drabb but this one has atmosphere and depth. With a "save current game feature" you don't have the normal frustration of starting off again from the beginning. There's plenty of puzzles to solve along your way so it's not a game you'll soon tire of. The Curse is available from none other than Micro Discount at a very reasonable price of £6.95, a very good price when you consider the work here.

TYNE&WEAR
ATARI
USER GROUP

T.W.A.U.G.
P.O.BOX No.8
WALLSEND
TYNE & WEAR
NE28 6DQ

## BANK - BANG

Review by Tommy Wood



BANK-BANG is one of the new Atari Classic games from GUR SOFT to be released by MICRO DISCOUNT (Derek Fern) at this years A.M.S.T. on November 13th at STAFFORD.

The town's bank manager has received information over the wire from the U.S. Marshal that a band of desperate bandits have planed to rob his bank sometime over the next few days. The Marshal is busy chasing the Jame's boys so he suggested apointing a armed rooting shooting lead spitting deputy to defend the bank.

Nobody wanted the job so all the sober cowpokes names were placed into a ten gallon hat and guess what pardoner ???? Yep it's up to you.

Bank - Bang is a game to test your quick draw reflexes to the full.

On the main playing area you will see three doors, the doors open and close at random. When the doors open you will see either a saloon girl a customer or a bandit with a gun!. You shoot only the bad guys, accidentally shooting any other and you loose one of your three lives.

For shooting the bad guys or girls, you gain points and dollars, you must fill the three windows at the top of the screen with dollars before moving on to the next set of doors. You can scroll across each set of doors by pressing the fire button and moving your joystick left or right, but only when all doors are closed.

On day two, you face "Jake" the gun-fighter, because you are one of the good guys you never fire first you must wait until he goes for his gun. Shoot to early and you loose a life.

When you have survived three days you face the night and a new foe "Pete" the Mexican bandit. Survive this far and you are now challenged to a Shoot out against 3 desperados for bonus points.

Day four brings "Bart" the super-fast gambler.

As you progress through the week the number of bad guys increase and the doors open and close at a much faster rate.

BANK - BANG continued

At each day you start with three lives, if you survive you carry the lives forward to the next day. At the end of your lives you can save your high score by moving your joystick on to each letter of your name and pressing the fire button, move to the square icon to enter blank spaces. The new high score table is then saved to your disk..

GOOD SHOOTING!!!

This game is not as easy as it looks. While loading you are presented with the scene outside the local bank. On pressing the fire button you are inside the bank looking at three doors. A catchy tune plays all the way through this game which can be annoying to some players, (ie.Me.) I could not find a way to switch it off. Even so the graphics are very good and I could become addicted to this game. Worth having in any Game Players collection.

# BRUNDLES

### Review by Max Gerum

This game is one of the new releases by MICRO DISCOUNT. I believe it is a two disk game and what I can see by the screen display it has high resolution graphics.

When you first boot the disk you get the title screen with options for choosing levels by entering a code. You press the Select key to enter the code. Pressing the Option key will take you to the score screen, and the Start key is of course to begin play. Once you have selected your options you go to the customizer screen here you will find all the games controllers and there are six of these, you can also select which port you want to plug the chosen controller in. Use the cursor keys to highlight your choice of the controller and press RETURN you then do the same to switch between port #1 and #2. After you've pressed the return key you will be presented with another screen where you have the opportunity to choose the memory your computer supports. The larger the memory the more screen levels will be loaded into your RAMdisk, there are around one hundred levels, yes 100 screen levels.

Here are the controllers you can choose from:
ATARI ST MOUSE
COMMODORE AMIGA MOUSE
SUNCON ANIMATION STATION
ATARI TOUCH TABLET
ATARI CX85
and finally but not least the JOYSTICK

The memories you can choose from is "none, 64K, 128K, 192K, and 256K. After you have made your selection and pressed return the first level will load.

I cannot comment on the payability of the game, the disk I am reviewing is a non-payable demonstration disk. The graphics look very good and the music it is playing throughout is a very catchy tune, the Select key switches the music on/off. If you like a good challenge then this game is for you. The price will be around £15.00, how many games can you find with this many levels.

# DISK CONTENTS

Side A

SPACE WOOLF: This was the first programme Richard Gore wrote when he was 14 years old. It is in Basic and you shouldn't have any problems getting the hang of it.

MAGIC SQUARE: Just as the title implies, squares with numbers to shuffle about to get a certain total amount. There are different levels and the instructions are displayed on the screen when loaded. It is a Basic programme and is fun to play.

BOMBER: It is something similar to the game that was in the old Atari User "Bomb Run". This is a Helicopter and you must knock down the towers with bombs.

ROBO-VENTURE: I am afraid I really cannot tell you much about this game, I haven't had the time to check this out. It is in Basic and should be easy to follow.

FOLDERLABEL: If you have a ring folder then this label printer is for you. It prints onto long labels and prints the text sideways. Ideal to stick onto the fold of the folder or to slip under the clear plastic cover.

Side B

This side is setup as AUTORUN.SYS,
EMPIRE: is a two player Strategy game. It is not easy to play, so you might find it challenging.

ARCed file: This ARCed file contains more printer driver screen dumps, it is for the XEROX 40-20 inkjet printer, by Ralph Bradley. The doc file is included in the arced file and can be printed out after unarcing, the extender for the docume ".MO2", so check all files with this extender.

# GERMANY's ATARI 8-BIT SHOW

report by Derek Fern

At last I have found a few minutes to jot down my comments and views after a very hectic weekend. Regular readers of Atari User will already know the original plan was to hire a 56 seater coach and take many Atari user's. Unfortunately like most plans the idea was good but the enthusiasm from 8 bit users was sadly lacking only 27 people phoned to book places on the trip, the cost of taking only 27 made it very expensive so the trip regrettably was cancelled. So yours truly and three other dedicated atari 8 bit enthusiasts travelled to Hanau a small town close to Frankfurt in Germany. Driving in Germany is like taking part in a Formula 1 race nobody travels at less than 100 mph !! 1200 miles on the wrong side of the road, at speeds averaging 85 to 90 mph even then the only things I managed to overtake was 3 dust carts and a battered old 2CV.

Hanau is over 600 miles away so I decided to spread the trip over 3 days with two over night stops one at Brussels on the outward leg and Maastricht "that well known treaty town" on the return. To cut costs we chose to stay a Youth hostels. Its certainly not the kit bag and sweaty sock brigade I remember from my scouting days continental youth hostels offer what can only be described as 5 star luxury, accommodation plus a continental breakfast for around £8:50 per night fantastic value for money.
As for the show, it was smaller than I expected I counted only 13 exhibitors located in two large ground floor rooms.
Amazingly the show did not have the massive discount bargain offers you would expect to see at similar U.K shows like A.M.S.
Kemal Ezcan the exhibition organiser from KE-SOFT greeted us with "Sorry no Brundles yet " but he has supplied a demo disk showing 4 of the completed screens. Also present A.N.G. Software and "Freddy" of MEGA MAG fame from Holland. Apart from the odd one or two new Polish games most stands displayed what can only described as standard items Rom cartridges and disk software which have been available in the U.K. for sometime.
However some exhibitors had some very new and original ideas unfortunately not all of the new items are yet completed and several exhibitors only had brief demos of the intended finished product, it was a bit like tomorrows world you can see it but can't get it.
But it was this new developement part of the show that for me made the whole trip worthwhile.
Over the next few months I will be in touch with all of these developers to try and encourage them to complete some of their brilliant ideas so here's a few details to wet your appetite. I have listed each company name for reference and I would like to thank them for their cooperation and patience in explaining their products and providing information on their products to a none German speaking Englishman.

H.S. COMPUTER: Demonstrated a range of interfaces and software that will link your computer to drive electric motors and servos for robotics and all sorts of mechanical devices.
They had a robot arm that could move in all directions and pick up items, a pen plotter that could be used to draw graphs, monitor and record via sensors. ie. temperature variants over a given time span.
The main robotic bits seem to be constructed from Fisher Technic parts and controlled via the joystick ports on a standard 800XL.

GERMANY's ATARI SHOW continued

ABBUC MAGAZINE: The German Atari User group displayed some of their vast range of club software, they have a very good subscription only disk based club magazine unfortunately for us it is only available with German text. Wolfgang Burger their Director said he would might be willing to make the disk available to U.K users if enough people wanted, so its up to you to let me know if you want to see it.
Wolfgang also expressed his concern about some U.K. suppliers pirating and openly advertising in Atari User Magazine software for sale which is copyrighted by the ABBUC club in Germany. If this continues he will lodge a complaint with a European court and sue for copyright infringement !, I understand the penalties in Germany are fines up to 2,500 DM per pirate copy sold.

FRIDAY SOFTWARE PRODUCTIONS : Friday Soft was a new name to me, its a small 2 man company developing a brilliant new 16 bit digitised sound replay system. The days of the 7 seconds of digitised sound with the 2 bit systems cartridge are over ! Digitised sound files produced on Amiga. St's and PCs are converted and compressed into 64k or 128k files and then replayed on your humble 8 bit machine. After loading the main program disk called "Fampy V2.3" you simply load one of the sample files press return and sit back in amazement. The sound quality is great especially if you have your computer hooked up to you Hi Fi system via the monitor port, the playing time for a digitised 64k file is 3 minutes and 8 minutes for a 128k file. The program is a replay system only, the equipment required for file compression and conversion is expensive and complex. I intend to have FAMPY available soon in the U.K with a set of sound sample disks.

ARGS HARDWARE: New exhibitor very much into programming demos, and hardware development. They displayed one of the best LEMMINGS demo's I have seen, excellent sharp animation produced with a bit map images.
ARG kindly gave us a copy of this demo so you can see an example of their work at the A.M.S 7 Show.
They also have under development a RS 232 interface that plugs into the cartridge port, it carries on board a real time clock and has a Baud rate of 19,200 ideal for high speed modem work. Also under development a P.C interface for down loading text files from a P.C.

KLAUS PETERS ELEKTRONIC + SOFTWARE : Mainly hardware stand, Klaus is the boffin who designed the speedy and superspeedy disk drive upgrades, his latest development is 1meg upgrades for 800XL and 130XEs and the machine retains full compatibility with all current software priced at 170 DM.
Klaus has also developed a 512K ROM DISK, this unusual device plugs into the expansion bus on the 800XL it is a bare printed circuit board with 5 empty I/C sockets and one EPROM and a few support I/Cs.
The EPROM carries the main operating program and menu selection. You can simply blow your utility or game into a 256K EPROM insert the EPROM into one of the empty sockets then select the prom from the on screen menu. This gives you a virtual instant load speed of any program you select. Priced at 165DM for the XE version.

GERMANY's ATARI SHOW continued

A.M.C. VERLAG : Very interesting stand, A.M.C are the programmers of the disk game HERBERT 1 & 2. They had on offer lots of the standard Atari items but also one very special demo of a soon to be completed skiing game. What made this demo stand out if you'll excuse the pun was it was in 3D !! yes real jump of the screen and smack you in the mouth 3D. The programmer explained the game had been underdevelopment for 5 months and he expected it to take another 3/4 months before completion. The demo was to demonstrate the principle of 3D programming it consisted of a landscape scattered with wire frame moving shapes and a matchstick man walking around wire frame pyramids the screen was blurred to look at without the green/red cardboard specs but put them on and out the screen popped true 3 D. "Very impressive" can't wait to see the game. Price estimated to be 34DM. This demo can also be seen at the A.M.S. Show.

TOP MAGAZINE: Well stocked stand, plenty of software/hardware including their own excellent disk based magazine.

DRACHEN SOFTWARE Germany: Demonstrated a very unusual art package on a high resolution paper white monitor. The demo showed an 8 bit window scrolling across a very large canvass made up with high resolution ST clip art pictures, everyone on the stand was busy demonstrating so I passed by planning to return later in the day but unfortunately when I returned they had packed up and left so no more details available.
I have the companies address and I will obtain more details if any one is interested the program certainly looked interesting and to be of excellent quality.
Their was not enough time to talk to everyone and I missed POWERSOFT and M. REITERSHAN but I was certainly encouraged by the enthusiasm shown by the dedicated few who spend a considerable amount of time sweating over our beloved 8 bit.

You will be able to see all of the new software demos collected from the trip at Stafford at A.M.S 7 exhibition.
Kemal Ezcan from KE-SOFT Germany and John Maris and Freddy from A.N.G & Mega Magazine Holland will also be exhibiting as part of the Micro-Discount stand A.M.S 7 promises to be the bigger and better than ever so see you there.

# AMS 7

## AT BINGLEY HALL

## STAFFORD

## on Saturday 13th November

*We have a stand at the hall and we aim to demonstrate the BLACK BOX. If you are interested in the BB and want to find out more about it, then here's your chance. The Catalogue from CSS, the producers of the Black Box, will be available at our stand, it includes the prices for the Black Box and for all the other products available for the 8-Bit Atari.*

*All back issues of the newsletter will be available plus the latest issue #6. PD disks can be purchased they will be prepared while you wait, or you can place your order and pick it up after you had a look at the other stands.*

Hope to see you at the all

MICRO SHOW...